ENCYCLOPEDIA OF COMPUTER SCIENCE AND TECHNOLOGY

计算机科学技术百科全书

（第三版）

中国计算机学会 CCF

清华大学出版社

ENCYCLOPEDIA OF COMPUTER SCIENCE AND TECHNOLOGY

计算机科学技术百科全书

（第三版）

Third Edition

主　　编　张效祥
执行主编　徐家福

清 华 大 学 出 版 社
北　京

图书在版编目(CIP)数据

计算机科学技术百科全书/张效祥主编. —3版. —北京：清华大学出版社，2018
ISBN 978-7-302-49572-7

Ⅰ.①计…　Ⅱ.①张…　Ⅲ.①计算机技术－百科全书　Ⅳ.①TP3-61

中国版本图书馆CIP数据核字(2018)第027518号

责任编辑： 薛　慧　张兆琪
封面设计： 傅瑞学
责任校对： 王淑云
责任印制： 李红英

出版发行： 清华大学出版社
网　　址： http://www.tup.com.cn，　http://www.wqbook.com
地　　址： 北京清华大学学研大厦A座　　**邮　　编：** 100084
社 总 机： 010-62770175　　**邮　　购：** 010-62786544
投稿与读者服务： 010-62776969，　c-service@tup.tsinghua.edu.cn
质量反馈： 010-62772015，　zhiliang@tup.tsinghua.edu.cn
印 装 者： 北京雅昌艺术印刷有限公司
经　　销： 全国新华书店
开　　本： 185mm×260mm　　**印　张：** 94.25　　**彩　插：** 1　　**字　数：** 2918千字
版　　次： 1998年8月第1版　　2018年5月第3版　　**印　次：** 2018年5月第1次印刷
定　　价： 498.00元

产品编号：035279-01



本书第一版获第十二届中国图书奖

《计算机科学技术百科全书》(第三版)
主编、副主编及各分支主编、编委名单

主　　编　张效祥

执行主编　徐家福

副 主 编　杨芙清　董韫美　李三立　李国杰　赵沁平

1. 计算机科学理论

主　编　殷建平

编　委　李　祥　李　廉　李舟军　莫则尧　朱大铭　祝　恩

2. 计算机体系结构

主　编　郑纬民

编　委　钱德沛　刘志勇　金　海　过敏意　陈文光

3. 计算机软件

主　编　梅　宏

编　委　朱三元　郑国梁　施伯乐　王　珊
王立福　王怀民　张　健　周兴社

4. 计算机硬件

主　编　韩承德

编　委　唐志敏　舒继武　冯　丹　谢长生　肖利民　苏振泽

5. 计算机网络

主　编　李　星

编　委　李忠诚　张　蓓　龚　俭　汪为农　张　凌　马　严　徐明伟

6. 计算机应用技术

主　编　张福炎

编　委　彭群生　徐光佑　蔡莲红　吴泉源　孙志挥　黄宜华
肖田元　俞士汶　徐晓飞

7. 人工智能

主　编　石纯一

编　委　王　珏　刘大有　史忠植　黄厚宽　俞士汶　周志华
张长水　刘椿年

第 三 版 前 言

《计算机科学技术百科全书》(以下简称《全书》)第二版出版至今已 10 年有余,在这十多年的时间里,计算机科学技术迅速发展,又涌现出许多新的内容。为适应广大读者获得计算机科学技术领域更新的知识的需求,我们从 2010 年开始筹划、组织《全书》第三版的编撰工作。

进入 21 世纪后,由于计算机科学技术的快速迭代和市场的迅猛成长,计算机产业已经步入成熟期。计算机、计算机网络、计算机的应用成为绝大多数人生活的基本工具和重要组成部分。因而,计算机科学技术的内容也成为人们必须学习的基础知识,作为“百科全书”理应承担“无围墙大学”的教学责任,为此,在本次修订中着重对有关计算机最基本、最重要的概念再次进行了审改,力求完善、准确。在“总论”中的“基本概念”和“基本内容”部分,分别对“计算机”“计算机科学”“计算机技术”“计算机科学技术”“计算机体系工程”“计算机产业”“计算机科学理论”“计算机体系结构”“计算机软件”“计算机硬件”“计算机网络”“计算机应用技术”“人工智能”的定义均做了完整、严谨的论述。“总论”中还增加了近十年“计算机科学技术”和“计算机产业”发展的新内容。

在本次修订中,对二级框架未做大的改动,仍为“计算机科学理论”“计算机体系结构”“计算机软件”“计算机硬件”“计算机网络”“计算机应用技术”“人工智能”七个分支。与二版比较,只将“计算机组织与体系结构”更改为“计算机体系结构”,并撰写了释文。为使各分支更准确涵盖本学科领域知识,在分支之间有个别部分的调整。例如,将“管理信息系统”“电子商务系统”等计算机应用软件的条目调入“计算机软件”二级框架内,新建立了“计算机应用软件”四级框架。在各分支中,又删除了已现陈旧的条目;进一步归并了一些较细的条目;增加了反映新技术应用的如“计算机学科交叉新技术”“网络存储”“数据库应用技术”等系统条目。对反映新技术、新理论且已成熟形成固定概念的新知识主题则列为新的条目,如“云计算”“服务计算”等。网络安全问题日益突出,也增加了相关的条目。

修订后,全书共有 1509 条,其中,“计算机科学理论”179 条,“计算机体系结构”158 条,“计算机软件”394 条,“计算机硬件”181 条,“计算机网络”204 条,“计算机应用技术”267 条,“人工智能”126 条。

计算机科学技术发展迅猛,虽然我们不断在做《全书》的修订工作,力求跟上其发展

步伐,使《全书》的内容能展现和记录计算机科学技术的状况,但总有遗憾。不足之处望广大读者不吝赐教。

衷心感谢上百位作者在第三版修订中付出的心血以及清华大学出版社对出版本书的大力支持。

《计算机科学技术百科全书》编撰委员会

2015.12

第二版前言

《计算机科学技术百科全书》(以下简称《全书》)第一版自1998年出版以来颇受各界关注。它已成为我国计算机学术、教育、产业与应用等诸领域工作者和广大计算机爱好者常用的、得力的工具书。由于计算机科技发展迅猛,自第一版发行至今6年多的时间里,又有了许多新内容涌现。为适应广大读者及时获得计算机科技领域更新知识的需求,从2000年开始筹划,在第一版的基础上进行了增、删与改写,并对《全书》框架作了适当调整,历时4载有余,方完成了《全书》第二版的编撰出版工作。

计算机网络在推进我国国民经济发展与社会信息化的过程中有着十分重要的作用。近年来计算机网络技术发展十分迅速,内涵丰富,为此,在《全书》第二版中将其列为独立分支。在原有的"计算机科学理论"、"计算机组织与体系结构"(第二版改名"计算机组成与体系结构")、"计算机软件"、"计算机硬件"、"计算机应用技术"和"人工智能"6大分支基础上增加了"计算机网络"分支。第二版在各个分支中除了适当增加新条目外,还对第一版框架中一些过于细小条目进行了合并、改写与删节,对一些已现陈旧的条目则予删除,以使整个框架更加合理,更能反映计算机科技的新近发展。第二版对"计算机科学技术总论"也做了一些修改与补充。全书共收录条目1 381条,比第一版略有增加。其中,"计算机科学理论"172条、"计算机组成与体系结构"133条、"计算机软件"320条、"计算机硬件"162条、"计算机网络"245条、"计算机应用技术"244条、"人工智能"105条。

《全书》第二版力求对每个条目赋予简明而确切的定义以及较为明确而完整的内涵,取材于肯定成熟的知识,言必有据,确切可信。

《全书》第二版广大作者撰写条目备极艰辛,清华大学出版社各位编辑精心校勘,于此均表感谢。

对《全书》第二版不足之处,欢迎广大读者不吝赐教。

《计算机科学技术百科全书》编撰委员会

2004年12月

第一版前言

半个世纪以来，计算机科学技术以磅礴之势迅猛发展，它以非凡的渗透力与亲和力，深入人类活动的各个领域，对人类社会的进步与发展产生了巨大的影响。计算机应用于科学研究，大大增强了人类认识自然与开发、改造和利用自然的能力，促进了现代科学技术的发展；计算机应用于生产，大大提高了人类物质生产水平和社会生产率，促进了经济的发展；计算机应用于社会服务，大大扩大和改善了服务范围与质量，提高了工作效率，推动了社会进步；计算机应用于社会文化，为人类创造文化提供了现代化工具，改变了人们创造和传播文化的方式、方法和性质，大大扩展了人类文化活动的领域，丰富了文化的内容，提高了质量；计算机进入办公室、家庭和为个人所拥有，正改变着人们的工作方式和生活方式。计算机科学技术对一个国家在政治、经济、科技、文化、军事、国防等方面发展的催化作用和强化作用，都具有难以估量的意义。它已在世界范围内形成一种现代文化。计算机科学技术在即将来临的新世纪中，必然会成为人类的重要基础文化知识之一。《计算机科学技术百科全书》（以下简称《全书》）就是迎接时代的需要，为推动我国计算机教育的普及与深入，为提高全民族计算机科学技术文化素质，为促进计算机学术研究与产业发展而撰写的。

50 年来，计算机已发展为范畴宽广、内涵丰富的科学技术和规模恢宏的新兴产业。计算机与通信的融合和全球性联网，更赋予它未可限量的发展前景。《全书》力求涵盖计算机科学技术发展 50 年来的主要成就，广收博取，深入浅出，以通俗精练的文体，比较系统全面地介绍学科的基本知识，既适于各行各业广大读者查询，也可供计算机专业人员参阅，并作为向深广发展的桥梁与阶梯。

《全书》根据计算机学科的内在联系、相关程度与性质特点，划分为“计算机科学理论”、“计算机组织与体系结构”、“计算机软件”、“计算机硬件”、“计算机应用技术”和“人工智能”6 大分支，按 4 级框架，共设置 1 293 个条目 200 多万字。由于中文信息处理是我国及全球汉字通用地区计算机应用中的重要技术，特在“计算机应用技术”分支中，设置有关中文信息处理条目 80 余条，以供读者查阅。《全书》按照不同层次与内容涉及范围，将条目释文分为大、中、小 3 类。在释文中有一定释义的常用名词术语还择要列作“主题词”者共约1 031个，与条目一起编入内容索引中，以利查阅。《全书》“总论”全面总览了计算机科学技术的内涵与对人类社会发展的巨大作用与深远意义，以引导读者全面、科学地认识计算机科学技术。《全书》设附录 3 种。

《全书》的撰写，着意于以简练、概括的笔触给每个条目以确切的定义和明确、完整的内容，取材于肯定成熟的知识，言必有据，确切可信。

《全书》在中国计算机学会、我国多所著名院校和计算技术研究所以及清华大学出版社等的支持和直接参与下，于1993年6月着手开展工作，至1996年12月完成全部编审，历时3年有余。有分布于全国的约400名专家、教授参与了撰写和审稿。《全书》是我国计算机学术界一部集体智慧的巨著，实现了大家多年来的共同愿望。

对《全书》不足之处，欢迎广大读者不吝赐教。

《计算机科学技术百科全书》编撰委员会

1997年2月

凡　例

1. 本书是全面、系统、科学地介绍计算机科学技术知识的专业性百科全书。全书按7个学科分支（计算机科学理论、计算机组成与体系结构、计算机软件、计算机硬件、计算机网络、计算机应用技术、人工智能），4级框架的层次设置了1 381个条目。它们构成了计算机科学技术完整的知识主题系统。

2. 条目分类目录由条目的题名组成，它们按学科分支的框架层次编排，体现了计算机科学技术所涵盖知识的内在联系、相关程度和性质特点。可以说，它是全部条目的系统表。

目录中有的条题名没有注明页码，表明该条题没有释文。设置这些条题的目的是为了完整地表示一个学科分支的科学体系。例如，计算机硬件分支中的"计算机逻辑部件"条题名。

目录中有的条题名会在两个分支内出现，说明它在两个学科分支体系中都有其作用。例如"自然语言处理"条题名在计算机应用技术分支和人工智能分支中均存在。

3. 本书条目按条题名汉语拼音字母顺序排列。第一字同音时，按其音调的四声顺序排列；同音同调时依次按笔画多少和笔顺排列；如完全相同，则按第二字，余类推。非汉字开头的条题名排在汉字条题名之后。依次为英文字母、希腊字母和阿拉伯数字开头的条题名，它们分别按字母顺序和数的顺序排列。

4. 全书的"参见"体系有两类情况：一类是两个条目释文内容相同但有两种条题名，则在一个条题下有释文，对另一个条题则注明"参见×××"。例如，"静止图像的压缩编码标准"条题内注明了"参见**图像的压缩编码**"。另一类是在一个条目的释文中阐述的部分内容另有专条论述或与其有关，则以"参见"方式表示。若参见的条题名在释文中出现，则该条题名用黑体字排出；若条题名在释文中不出现，则加括号，并在括号里注明参见的条题名，同时用黑体字排出。

5. 释文内用魏碑字体排出的主题词（如*微程序*）是未被本书列为条目而在文中有定性叙述或较多阐释的知识主题。

6. 本书的检索系统有：条目汉语音序索引、条目外文索引和内容索引。

内容索引中包含了全部条题名和释文内的主题词。这些主题词用魏碑字体标示，有的层次标题又是主题词，则其字体不变。

7. 书末有3个附录。附录Ⅰ为书中出现的部分英文科技名词的缩略语。附录Ⅱ为计算机及相关学科科技期刊名录，包括中国期刊和外国期刊。附录Ⅲ为计算机及相关学科学术团体名录。

8. 书中科学技术名词采用全国科学技术名词审定委员会公布的名称。尚未公布的则采用本专业中惯用的名称。

书中采用由国家技术监督局发布的《量和单位》中所规定的物理量及其单位。对一些非法定计量单位则依惯例全书统一,并给出与法定计量单位的换算关系。对计算机技术中常用的单位做到全书统一,如用 B 表示 byte,b 表示 bit,KB 表示 1 024 byte。

外国人名已有通用中文译名者,如牛顿、傅里叶,按译名写出;其余的在文中第一次出现时加括号写出原文。对专业性较强的条目,其中外国人名不予翻译。

目　录

计算机科学技术总论

本文包含基本概念、巨大作用、发展历程、基本内容、计算机产业,以及发展展望六部分。

基本概念

计算机科学技术是以计算机为研究对象的科学技术。

计算机是一种现代化的信息处理工具。信息是对数据所赋予的含义,数据是对象的表示,无含义。处理可理解为变换。可用之物为器具,作用鲜明且可导致可用结果的器具为工具。现代化可理解为反映现代科技水平。联系到计算机,可理解为20世纪30年代以前为近代,20世纪30年代起为现代。计算机对信息进行处理,并提供处理结果。

科学是旨在观察现象、发现规律、探求真理的系统化知识。知识可理解为可信的一组事实与一组规则,有人将知识定义为一个三元组(事实集,规则集,置信度),其中置信度为含于[0,1]区间中的某一实数,它反映相应事实集与规则集的可信程度。系统化则指内容可以构成系统。计算机科学是以计算机为研究对象的科学。

技术有两层含义。一为实体含义,意指科学之理论、原则、方法在某一领域中的应用。二为学科含义,意指以实体含义之技术为研究对象的学科。计算机技术是以计算机为研究对象的技术。

计算机科学技术是计算机科学与计算机技术的统称,它是研究计算机的设计、制造,以及利用计算机进行信息获取、表示、储存、处理、控制等的理论、原则、方法和技术(这里技术英文为technique,科学技术中之技术为technology,二者含义有别,前者反映个体,后者反映总体,汉语未加区别,其理解须视上下文而定)的一门学科。

为了深入理解计算机科学技术,除了上述之计算机、计算机科学、计算机技术外,尚须了解与之密切相关之“计算机工程”与“计算机产业”。

工程亦有两层含义。一为实体含义,二为学科含义。前者意指科学技术之理论、原则、方法、技术(techniques)在某一项目上的应用。后者则指以实体含义之工程为研究对象的学科。计算机工程则是以计算机为研究对象的工程。

产业是关于产品的设计、制造、生产、销售,以及售后服务的所有机构的总称。计算机产业是以计算机为产品的产业。

计算机、计算机科学技术、计算机工程、计算机产业之关系是:

计算机是计算机科学技术、计算机工程的研究对象,是计算机产业的产品。

计算机科学技术、计算机工程是计算机和计算机产业的发展源泉，计算机产业是计算机、计算机科学技术、计算机工程的发展依托。四者又同时限定和制约于社会发展、经济发展，以及相关学科的发展。

巨 大 作 用

计算机是20世纪40年代人类的伟大创造。它对人类社会的进步与发展作用巨大，影响深远。

1. 开拓了人类认识自然、改造自然的新资源

人类最早认识和开发的是物质资源，把它转化成材料，制作出简单的工具，从事个体、家庭或小作坊的生产，这种生产方式导致生产率低下，以致社会经济发展缓慢，形成几千年的农业社会自给自足的自然经济。18世纪以蒸汽机发明为标志的产业革命兴起，开启了能量资源的开发和利用，把它转化为动力，制造出各种自动的机器作为生产工具，有效延伸了人的体能，劳动生产率显著提高，使人类进入大规模生产的工业化时代，形成以商品生产与交换为标志的市场经济。工业化为人类创造了巨大的财富，促进了社会经济的繁荣与发展，改变了社会的结构，但同时也带来了非再生物质资源和能量资源的大量消耗与浪费。人类发现信息这一战略资源，还是近几十年的事。现代科学技术的进步，特别是计算机的出现，使人类从此有了自动化、信息化和一定智能化的强大工具，以开发利用信息资源，把它转化为知识产品，促使物质生产水平和社会劳动生产率空前提高，开创了信息时代的新纪元。以计算机为核心对信息资源的开发和利用，使物质资源和能量资源的效益得以更加充分、高效地发挥，人们能以合适的物质和能量创造出高质量产品，其增值来源于信息和知识。计算机与通信的融合，建立大量信息网络和大规模高速互联网，必将深刻影响人类的生产方式与生活方式，形成以信息和知识产品为特征的“信息经济”和“知识经济”。计算机的出现，使人们在物质和能量两大战略资源外，开发和利用了“信息”这一新的战略资源，开拓了人类认识自然、改造自然的新资源。

2. 增添了人类发展科学技术的新手段

长期以来，人类发展科学技术依靠两大传统手段，即理论与实验。这两种手段起过并继续起着基本作用。而计算机的出现，由于其自动、高速进行大量运算的能力和计算的精确性，致使过去科学家穷毕生精力无法办到的事，如今在短短几小时，甚至几分钟内即可变成现实，并能获得单纯依靠理论与实验难以得到的结果。从而，一方面，使传统物理学、化学、生物学等基础科学的研究进入了新的境界，出现了计算物理学、计算化学、计算生物学、计算力学等新兴学科；另一方面，在诸如电机工程、土木工程、建筑工程、化学工程、航空工程、材料工程等工程性学科的研究中，由于利用了计算机这一现代化信息处理工具以及计算机科学技术的研究成果，更新了研究手段，加速了它们的发展。同时，由于计算机科学技术与其他学科的融合，出现了人工智能、计算机图形学、虚拟现实等交叉性学科。此外，计算机用于自然资源开发、重大工程建设与环境保护等方面，正在起着越来越大的

作用。计算机已在航空航天、资源勘探、大范围中长期天气预报、材料、遗传工程、核能利用、尖端武器设计等众多领域,取得了重大经济效益和社会效益。随着计算机应用的不断拓广与深入,以及计算机科学技术的不断发展,必将出现更多的新兴交叉性学科。计算机和计算机科学技术的出现,在理论与实验两大传统手段外,又增添了一种人类发展科学技术的新手段,即计算手段。

3. 提供了人类创造文化的新工具

文化是人的行为以及体现在思想、言语、行动、制作中的成果的总汇,是人类创造的社会精神财富和物质财富的总和。计算机用于辅助教育,丰富了教育手段与方法。计算机辅助教育以生动的画面和动画图形来描述数学、物理、化学、历史、地理与语文等学科内容,寓教育于娱乐,以形象补充文字,提高了学习者的积极性。计算机辅助教育通过学习者与计算机之间的交互活动,使学习者能自主探索,按需学习,从而培养了学习者的创造思维和学习的主动性,收到传统教育方法难以收到的效果。以计算机为核心的电子照排系统,从文稿起草、编辑、版面编排,到制版印刷,连续完成一系列工序流程,大大提高了文化传播的能力与水平。随着电子印刷的推广,大量图文资料进入计算机硬盘、光盘等存储媒体,自然地引发了电子图书和数字图书馆的出现,几十卷的巨著存入光盘,既便于携带、查阅,又大幅度降低了出版成本。多媒体技术和超文本结构的引入,更将使电子图书、电子报章成为文化传播的手段。计算机进入美术、影视等领域已成现实。用计算机创作的编织、刺绣、服装、地毯、壁纸、工业造型与动画等已进入市场。在计算机上直接完成乐谱制作,用计算机设计舞蹈表演和人体动作等,已引起音乐家、舞蹈家和体操教练等的重视。机器翻译与语言文字识别等技术的进展,将在国际合作和科技文化交流等方面发挥重大作用。在各类学校中,计算机都列为必修课程。社会上各行各业的工作人员由于学会使用计算机,其工作成果的数量、质量与工作效率均大大提高。从而,计算机及其使用已成为人类必需的文化内容,计算机已成为与语文和数学同等重要的基础知识。计算机的出现,为人类创造文化提供了新的现代化工具。它改变了人们创造文化的活动方式、方法和性质;拓宽了文化活动的领域;丰富了文化的内容;提高了质量;革新了传播手段;改善了学习条件;增强了传播能力,使之达到前所未有的水平。

4. 引起了人类的工作方式与生活方式的变化

计算机进入办公室、家庭和个人之手,使人类的工作方式与生活方式经历着巨大变化。社会与经济的发展使各类社会组织如政府机关、企业事业部门、金融商业机构、社会团体等的业务信息急剧增长,决策处理科学化和时效性要求大大提高,传统的工作方式与方法已难以保证质量要求和决策水平。计算机技术、通信技术与各种办公设备相结合,使人类的工作方式与方法产生了巨大变革。人们用计算机进行文字处理;用电子报表、图形、图像、声音等多种媒体来表示工作中复杂、生动的实际情况;用电子邮件保证部门间信息传递的及时、方便与可靠;电子会议改进了会议方式,减少了会务工作,提高了会议效率。计算机、通信网络与各种计算机信息系统相结合,大大增强了人们掌握工作全局情况和综合分析判断的能力,有效地提高了决策、经营和管理水平。计算机进入家庭给家庭生

活带来巨大变化。人们借助计算机管理家庭日常事务;对家用设备如照明、煤气、空调、电源,以及门户、烟火安全等进行监控;计算机及其网络技术应用于商务活动,实现电子商务;应用于政务活动,实现电子政务;应用于医疗业务,实现远程医疗;应用于教育培训,实现远程教学;计算机与电视、电话相结合,可获得电视游戏、电视点播等服务;通过计算机网络,实现在家办公。笔记本计算机与智能手机的发展更使计算机成为易于携带的便携式计算工具,并通过通信网络为实现不拘地域、空间均能开展工作提供了条件。总之,计算机、计算机网络等给人类的工作方式与生活方式带来深刻变化,并由此步人信息化社会。

发 展 历 程

1. 国际

(1) **电子计算机的诞生**　用于计算的机器可追溯到 17 世纪,那时欧洲的一些数学家就设计制造出纯机械式的数字运算机器。著名的有 1642 年法国数学家 B. Pascal 制成的十进制加法器,1673 年德国数学家 C. N. Leibniz 研制的进行十进制数乘、除运算的计算器。在此基础上英国数学家 C. Babbage 于 1822 年研制成可以运转的差分机模型,1834 年他又设计了一种程序控制的通用分析机,但限于当时的技术条件,未能实现。在 Babbage 分析机之后的几十年中,数字式计算机的研究出现了停滞,但有一批物理学家用物理方法探求计算工具的新途径,兴起了模拟计算机的研制。模拟计算机借助连续物理量运算解算问题,用物理过程来模拟数学方程的解算过程。直到 20 世纪 30 年代模拟计算机仍受重视,但其专用性、低精度、可靠性与稳定性较差等弱点限制了它的推广。另有一类计算机曾在电子计算机出现之前起过重要作用,即 19 世纪末叶由统计工作者 H. Hollerith 等人创造的高级分类统计机。它以机电相结合的结构,采用穿孔卡片作为数据载体,完成分类、统计、制表等一系列计算操作过程。这类机器在 20 世纪 40 年代后逐渐被淘汰。最早采用电气元件研制计算机的是德国工程师 K. Zuse,他于 1941 年完成全继电器式通用计算机 Z-3。其他著名的继电器式计算机尚有由 H. Aiken 于 1944 年完成的 MARK-Ⅰ及 1947 年完成的 MARK-Ⅱ。科学技术的进步,特别是电子学的迅速发展和第二次世界大战对先进计算工具的迫切需求,为现代电子计算机的诞生奠定了社会与技术基础。首先采用电子技术实现的数字计算机为 1946 年 2 月美国宾夕法尼亚大学莫尔学院制成的 ENIAC(Electronic Numerical Integrator and Calculator),它含有 18000 个真空管,运算速度达到当时继电器式计算机的 1000 倍,但它没有采用二进制操作和存储程序控制,未具备现代电子计算机的主要特征。1945 年 3 月,J. von Neumann 领导的小组发表了二进制的程序储存式的电子离散变量自动计算机 EDVAC(Electronic Discrete Variable Automatic Computer)方案,1945 年 7 月,J. von Neumann 等人又提出更为完善的设计报告,宣告了现代计算机结构思想的诞生。但由于种种原因,直到 1951 年 EDVAC 才告完成。而英国剑桥大学的 M. V. Wilkes 在 EDVAC 方案的启发下于 1949 年制成的 EDSAC(Electronic Delay Storage Automatic Calculator)成为世界上第一台程序储存式的现代电子计算机。

（2）**器件更新作为计算机划代的标志** 计算机硬件的发展受到电子开关器件的极大影响。为此，器件更新被作为计算机技术进步划代的标志。第一代为电子管计算机（从20世纪40年代中期到50年代末期）。除了前述的ENIAC和EDSAC外，最具代表性的计算机尚有1951年的UNIVAC-Ⅰ和1956年的IBM 704等。1951年由J. P. Eckert和J. Mauchly主持设计的UNIVAC-Ⅱ是美国批量生产的第一台电子管商用计算机。为军事需要而研制的大型计算机则有1954年的NORC等。电子管计算机体积大、功耗大、故障率高，运算速度只在每秒一两万次左右。第二代为晶体管计算机（从20世纪50年代中后期到60年代中期），其主要特征是采用晶体管作为开关元件。和第一代的电子管计算机相比，第二代晶体管计算机具有体积小、可靠性高、功耗低、运算速度快（可达每秒执行百万条指令）等优点。最初出现的晶体管计算机为1956年美国军用的Leprechan，而美国麻省理工学院于1957年完成的TX-2对晶体管计算机的发展起了重要作用。这一代计算机的产品主要有IBM 7040、IBM 7070、IBM 7090等。小型计算机则有IBM 1401。主要的大型计算机有UNIVAC-LARC、IBM Stretch以及CDC 6600等。第三代计算机（从20世纪60年代中后期到70年代初中期）以小、中规模集成电路作为基础器件，这是微电子与计算机技术相结合的一大突破，致使可以廉价构作运算速度快、容量大、可靠性高、体积小、功耗小的各类计算机，其中有代表性的是IBM 360系列与PDP-11。第四代计算机（20世纪70年代中后期以来）的主要特征是普遍采用大规模与超大规模集成电路（LSI与VLSI）技术，从而导致计算机硬件价格急剧下降，机器的性能价格比迅速提高。

（3）**计算机应用方式的发展** 在计算机出现初期，所处理的大都是科学计算和工程计算问题，计算量大而数据量相对较少，主要采用批量处理方式。20世纪50年代后期，企业应用逐渐开展，数据处理问题日益增多，这类问题的数据量大，输入输出频繁，计算量相对较小，致使运算部件经常处于空闲状态。为使价格昂贵的计算机资源得以充分利用，以提高计算机系统的实际使用功效，出现了分时处理方式与交互作用方式。70年代微处理器的出现与80年代微型计算机（又称个人计算机PC）的大发展，使计算机得以进入各行各业、家庭和个人之手，大大加速了计算机的普及应用，出现了所谓个人计算方式。90年代以来，计算机网络蓬勃发展，大量计算机联入不同规模的网中，大大扩展和加速了信息的流通，增强了社会的协调与合作能力，使计算机的应用方式向分布式和网络式发展。

（4）**计算机产品的发展** 计算机发展初期，主要针对具体应用需求研制机器，因此，型号多而产量少。有一定批量的工业生产始于20世纪50年代前期。随着应用和计算机工业的发展，人们注意到计算机产品继承性的重要性。50年代后期出现了具有一定兼容关系的计算机系族，IBM 700、IBM 7000系族为其代表。在这一时期还因为工业、商业、金融业等对数据处理应用的需求，促使小型计算机如IBM 1401及PDP-8等的发展。1964年4月IBM公司发布IBM 360系列，对计算机的普及和大规模工业生产产生了重大影响。IBM 360以统一的体系结构、操作系统、输入输出接口，以及科学计算、数据处理、实时控制等广阔的应用方面，达到大、中、小型计算机之间的兼容，实现了系统的通用化、系列化与标准化，成为计算机发展中的重要策略。CDC、UNIVAC、Burrough等公司也都相继推出了系列化产品。系列机大量节约了后继机种的开发成本，缩短了开发周期。尤为重要的是，保护了用户的软件资源积累。20世纪70年代初，Intel 4004芯片研制成功，为80年代

微型计算机的大发展奠定了基础,掀起了计算机普及的浪潮。特别是 80 年代后期 RISC 芯片的出现,使芯片运行速度大大提高。微处理器的集成度和性能按照摩尔定律呈指数级增长,导致终端计算机的小型化,从台式机、手持笔记本计算机发展到平板电脑和智能手机。另一方面,大、中、小型计算机都向服务器方向发展,性能不断提高。1970 年前后,相继出现了 CDC7600,STAR100,ASC 等巨型计算机,其主要特点是,速度快,并行处理能力强。1976 年 Cray 公司推出了 Cray-1 向量巨型机(这种巨型机又称为超级计算机,后来又统称为高性能计算机)。为了满足各行各业对计算能力的强烈需求,20 世纪 80 年代以来,并行处理成为研究的热点。特别是,使用数以千计的微处理器组成的并行处理系统,其峰值运算速度已达每秒几十万亿次。

近三十年来,全球计算机产品异彩纷呈,例如,Intel 公司推出的奔腾系统(Pentium 60,66……),Apple 公司的 PowerBuk64 系列,IBM 公司的 64 位处理器 Power PC 910。2014 年全球最快的超级计算机“天河二号”,其计算峰值达每秒 5.5 亿亿次浮点运算。从上世纪末迄今,以 IBM,惠普,戴尔,联想等公司为代表之计算机企业蓬勃发展,先后推出了一大批迎合市场需求之高性能 PC 产品,行业进入了寡头竞争时代,同时,人们也更加注意对纳米技术、量子计算、神经元计算等新兴技术的研究,旨在研制出更加微型化、网络化和智能化的计算机产品。

(5) **计算机软件的发展**　计算机软件的发展受到应用和硬件发展的推动和制约。反之,软件的发展也推动了应用和硬件的发展。软件的发展经历了如下阶段:从第一台计算机上的第一个可用的程序开始到实用的高级程序设计语言出现以前为第一阶段(20 世纪 40 年代中期到 50 年代中期)。如前所述,在计算机发展初期,应用领域较窄,主要是科学计算与工程计算。处理对象是数值信息。编制程序所用的工具是低级语言(机器语言和汇编语言)。程序的设计和编制工作采用个体工作方式,强调编程技巧。研究对象是顺序程序。这一阶段主要研究科学计算与工程计算程序、服务性程序和程序库。当时人们对和程序有关的文档的重要性尚认识不足,重点考虑程序本身。那时虽尚未出现“软件”一词,但毕竟由于程序是软件的主体,从发展的连续性来看,仍应将其列为第一阶段。从实用的高级程序设计语言出现以后到软件工程提出以前为第二阶段(20 世纪 50 年代中期到 60 年代后期)。虽然早在 1951 年瑞士学者 H. Rutishauser 就提出设计高级语言及其翻译程序,但直到 1956 年在 J. Backus 领导下,才就 IBM 704 机器研制出第一个实用的高级语言 FORTRAN 及其翻译程序。此后,相继又有多种高级语言问世,著称者有 ALGOL 60,COBOL,SIMULA,ALGOL 68,PASCAL 等,致使设计和编制程序的功效显著提高。为了充分利用系统资源,出现了操作系统(如 IBM 360 操作系统)。为了适应大量数据处理问题的需要,研制了数据库及其管理系统。在 20 世纪 50 年代后期人们逐渐认识到和程序有关的文档的重要性,因此到了 60 年代初期,出现了“软件”一词,融程序及其有关文档为一体。这时,软件的复杂程度迅速提高,研制周期变长,正确性难以保证,可靠性问题相当突出。到了 60 年代中期,出现了人们难以控制软件开发的局面,即所谓软件危机。为了解决这一危机,人们进行了以下三方面的工作:第一,提出结构程序设计方法;第二,提出用工程方法开发软件;第三,从理论上探讨程序正确性和软件可靠性问题。这一阶段的研究对象增加了并发程序,着重研究高级程序设计语言、编译程序、操作系统以及各种

应用软件。计算机系统的处理能力得到提高,设计与编制程序的工作方式逐步转向合作方式。从软件工程提出迄今为第三阶段(20 世纪 60 年代后期以来)。由于大型软件的开发是一项工程性任务,采用个体或合作方式不仅效率低、产品可靠性差,而且很难完成,只有采用工程方法才能适应。从而在 1968 年的大西洋公约学术会议上提出了"软件工程"的概念。四十多年来,软件领域工作的主要特点是:第一,随着应用领域不断拓广,出现了嵌入式应用及其软件;为了适应计算机网络的需要,出现了网络软件;随着微型计算机的推广,分布式应用和分布式软件得到快速发展。第二,软件工程发展迅速,开发方式逐步由个体合作方式转向工程生产方式。除了开发各类工具与环境,用以支撑软件的开发、运行与维护外,还有一些实验性的软件自动化系统。第三,致力研究软件体系结构、基于构件的软件开发、起重要支撑作用的中间件、平台软件以及软件过程本身,研究各种软件开发风范与模型。第四,除了软件传统技术继续发展外,人们着重研究以智能化、自动化、集成化、并行化、开放化以及自然化为标志的软件开发新技术。第五,致力研究对象技术与主体技术。第六,注意研究软件理论,特别是软件开发过程的本质。第七,随着大数据、云计算以及"软件定义一切"理念的出现,软件技术将迎来大发展的时期。

2. 国内

(1) **中国古代的贡献** 中国古代在计算理论与计算工具方面贡献突出。主要发明有五:一为二进制的位。其表示符号为"爻"。爻分阴爻和阳爻。阴爻对应 0,阳爻对应 1,易经中的八卦和六十四卦分别源于 3 个爻和 6 个爻的集合。德国数学家 Leibniz 曾说:"伏羲在其推演的八卦中使用了二进制算术"。二为十进制记数系统。据殷墟甲骨文和周代青铜器上的铭文记载,十万以内的自然数可由 1 ~ 9 的 9 个符号和表示十、百、千、万位值的 4 个符号表示。较当时巴比伦和古埃及的记数制更为科学。中国早就把零当作数。公元 1 世纪的《九章算术》中已阐明了负数的运算规则,印度在公元 7 世纪才提到负数,欧洲到 17 世纪才有论述负数的著作。三为筹算。东周末年,出现了算筹体记数法,筹算利用算筹作为运算工具,春秋战国时期已广泛使用,它对中国古代社会的发展起了重要作用。四为珠算。它以算盘为计算工具,在元代已广泛使用,明代传至日本、朝鲜等国。五为提花机,这是一种提花织物的纺织机械,要织的图案先做成"花本",用花本去控制织造过程。提花机是一种过程控制的纺织机械,秦汉时期已经出现,法国到 18 世纪才用穿孔卡片机控制织造过程。

(2) **中国计算机系统的研制** 早在 20 世纪 40 年代后期,中国即有在欧美国家的留学人员进行计算机的研究与开发,1952 年在华罗庚的倡导下,中科院数学研究所成立了计算机研究小组。

中国计算机事业创始于 20 世纪 50 年代中期。1956 年国家制定《1956—1967 年科学技术发展远景规划》,将"计算技术的建立"列为四项紧急措施之一。一面派人去苏联考察、学习,一面在国内开办训练班,积极培养人才,同时筹建中国科学院计算技术研究所。并以苏联科学院资料为蓝本,分别于 1958 年与 1959 年研制出我国最早期的计算机,即 103 小型数字计算机和 104 大型通用数字计算机。此后开始自主研制。1960 年由中国科学院计算技术研究所研制出小型电子管计算机 107 机,1964 年 5 月和 10 月由中国科学院

计算技术研究所和华东计算技术研究所分别研制出大型电子管计算机 119 机和 J-501 机。1965 年南京大学与华东计算技术研究所合作在 J-501 机上配制了 ALGOL 语言。中国科学院计算技术研究所在 119 机上配制了 BCY 语言。1965—1966 年间中国科学院计算技术研究所、哈尔滨军事工程学院、华北计算技术研究所、华东计算技术研究所等单位分别研制出晶体管计算机：109 乙机、441B 机、108 机和 X-2 机。此外，投入生产的还有 121 机和 112 机。从而中国进入了晶体管计算机的时代。这些机器一般都配有 ALGOL 或 FORTRAN 语言。FORTRAN 语言是由长沙工学院(即今之国防科技大学)于 1973 年首先在 441B 机上配制的。中国集成电路计算机的研究始于 1965 年。直到 1971 年，中国科学院计算技术研究所的 111 机和华北计算技术研究所的 112 机才基本研制成功。1973 年北京大学与北京有线电厂合作研制出百万次级的 150 机，华东计算技术研究所也研制出性能和 150 机相当的 655 机，并先后投入运行。这些机器都配有高级语言与管理程序。1973 年初，原第四机械工业部主持研制 100 系列与 200 系列计算机。前者和 NOVA 机兼容，清华大学负责研制的 130 机与 140 机批量生产千余台；后者指标和 IBM 360 相当，但和 IBM-360 不兼容，也生产若干台，并配有 14 个软件系统，其中包括三个操作系统(南京大学研制 XT-1，北京大学研制 XT-2，华北计算技术研究所研制 XT-3)，FORTRAN(北京有线电厂主要研制)，COBOL(南京大学主要研制)，BASIC(西安交通大学主要研制)，以及系统程序设计语言(南京大学研制)等。此外，中国科学院计算技术研究所研制成 757 向量机与 KJ 8920 大型机。国防科技大学先后于 1983 年及 1992 年研制成向量式巨型机银河Ⅰ和银河Ⅱ以及后来的银河Ⅲ、银河Ⅳ、以及近期的天河一号、天河二号等，它们都配有操作系统、高级语言编译程序等系统软件，这些机器对国防建设与国民经济建设均起了重要作用。另一方面，清华大学开发出中华学习机，生产十余万台。长城计算机公司与清华大学联合研制的 0520 机是我国最早的国产微型计算机。随着对微型计算机需求量的日益增加，我国计算机的装机量从 1978 年的 500 台猛增到 1990 年的 50 万台，1996 年的 500 万台，2003 年已生产微型计算机 3000 多万台，2012 年已达 3.5 亿台计算机得到更为广泛的普及应用。此外，国家智能计算机研究开发中心于上世纪 90 年代陆续研制成功曙光一号对称式多处理机和曙光 1000 大规模并行计算机等高性能计算机。20 多年来，国防科技大学、江南计算所，上海大学和中国科学院计算技术研究所(曙光公司)分别推出了银河、天河、神威、自强和曙光系列超级计算机，性能达到国际先进水平，天河二号计算机连续四年在世界 TOP500 高性能计算机中排名第一，曙光计算机连续六年超过 IBM 公司等国外产品，在国内 TOP100 高性能计算机的份额(台数)排名中保持第一，中国已成为世界上研制超级计算机的强国之一。

(3) **中国计算机的应用**　中国计算机应用的发展可分为如下三个阶段。20 世纪 50 年代末至 60 年代中为第一阶段。其特点是，所解问题多为科学计算与工程计算问题，诸如求代数方程的近似解，求线性代数方程组的数值解，以及求常微分方程组、偏微分方程组的数值解等，处理对象均为数值信息。应用领域涉及国防建设、气象数值预报、工程设计等。程序人员使用低级语言编制程序，单纯手工方式，有一些简单的标准程序库与服务性程序。60 年代中至 70 年代末为第二阶段。这时所解算的问题除了科学计算与工程计算问题外，增添了数据处理问题。如前所述，这类问题计算量相对较小，数据传输量却很

大,输入输出频繁,处理对象主要还是数值信息。应用领域除前述者外,还涉及各种企业、事业部门,应用面不断拓广,开发了不少管理信息系统。程序人员普遍使用高级语言,各类机器一般都配有 ALGOL,FORTRAN,COBOL,PASCAL 等语言,以及各种操作系统等系统软件,解题环境得到改善。培养了一批系统软件及应用软件开发人员。这一阶段利用计算机解题的水平显著提高。80 年代迄今为第三阶段。其主要特点是,处理对象除了数值信息外,增添了非数值信息。既有数值问题,也有逻辑问题,应用面大大拓广。所解问题涉及国防建设、国计民生、教育文化、安全保卫和娱乐健康等方面。计算机辅助技术用于辅助设计、辅助制造、辅助教育以至辅助软件工程。在软件开发过程中也尽量利用了计算机系统。软件技术与人工智能技术相结合,出现了一些富有特色的计算机辅助设计系统、专家系统以及图形、图像识别与处理系统,具有一定智能的软件工具以及若干实验性的软件自动化系统等。60 多年来,中国计算机应用的发展已逐步从面向专业人员朝面向非专业人员过渡,由处理单纯数值信息发展为既处理数值信息又处理非数值信息,并发展了包括语言、文字、图形、图像、声音等在内的多媒体应用。计算机的应用面和应用水平不断拓展和提高。

(4) **中文信息处理** 中文信息处理是我国与全球汉字通行国家、地区和汉字使用者在计算机应用中面临的重大问题。

中文和西文的差异较大。60 多年来,特别自 20 世纪 70 年代中期以来,我国在中文信息处理方面进行了大量的研究开发工作。从汉字属性分析研究、汉字键盘输入技术、汉字字模技术、汉字输出技术、汉字编码以及储存、检索、软件汉化到中文篇章识别、汉语言语识别、手写汉字识别、篇章理解与处理、机器翻译、电子照排、印刷出版、中文平台等方面,取得了一系列重大成果。

对汉字编码输入,人们从汉字本身体现的各种特点出发,提出了数百种方案,实际使用者有十多种。但汉字输入问题的完善解决,尚有许多探索研究工作要做。为了储存大量中文篇章,需解决信息压缩问题,为此,提出了基于理论而又颇富实效的压缩技术。中文信息检索方面异彩纷呈,提出了多种中文信息检索方法。关于软件汉化,也做出了很好的工作,如汉化 DOS、汉化 Unix 等。

中文篇章的识别、理解与处理比较困难。为此,必须进行词切分,从语法、语义与语用三方面联系起来考察。多年来,在汉语语法方面开展了大量研究工作。在研制具体汉语处理系统的同时,还从理论上探讨了汉语语法的形式化问题。用合适的形式体系来描述受限汉语的语法,取得了较大进展。同时对难度更大的汉语语义的形式化问题也进行了探索,但是,在汉语语用的研究上,工作尚少。此外,在与中文篇章的理解与处理密切相关的语料库方面,也进行了卓有成效的工作。

汉语言语识别的难度很大,既要考虑汉语的平、上、去、入四声,又要考虑到上、下文,目前有的系统已初步具有学习功能,经过对发音者的语音学习后,即可识别出同一发音者的言语,而且达到较高的正确率。对手写汉字的识别也很困难。首先要确定字体(如楷、宋、隶、篆、草体以及简、繁体等);其次要考虑到各人写法的可允许差别范围。必须从识别方案、特征抽取、识别算法等多方面探求解决方法。目前已出现一些实验性系统,在特定使用领域并加以若干限制的条件下,有的系统已臻实用。此外,如机器翻译、电子照排、

印刷出版等,我国均有出色的成果,并有相当影响的产品问世。近年来,还在将中文信息处理系统的硬件支撑与软件支撑联系起来统一考虑,在建立计算机系统的中文平台方面做出了不少成绩。

基 本 内 容

计算机科学技术的基本内容可概括为计算机科学理论、计算机体系结构、计算机软件、计算机硬件、计算机网络、计算机应用,以及人工智能 7 个领域。

1. 计算机科学理论

计算机科学理论意指计算机科学中之基本概念与基本原理所构成之体系,主要包括数值计算、离散数学、计算理论、程序理论四部分。数值计算意指以数值信息为计算对象的计算。讨论用于模拟物理过程或社会过程的各种算法的设计、分析和使用。早在 18 世纪与 19 世纪,高斯、牛顿、傅里叶等著名数学家就研究过数值计算方法,而计算机的诞生则大大促进了数值计算的发展。数值计算所涉及的内容颇多,如方程求根、数值逼近、数值微分、数值积分、线性代数方程组的数值解法,矩阵特征值计算,以及微分方程数值解法等。

离散数学泛指数学中讨论离散对象的分支。和连续数学不同,离散数学通常涉及整数系。由于数字计算机是离散机,离散数学之重要性不言而喻。通常认为离散数学包括集合论、图论、组合学、数理逻辑、抽象代数、线性代数、差分方程、离散概率论等学科。

计算理论主要包括算法、算法学、计算复杂性理论、可计算性理论、自动机理论、形式语言理论等。

程序理论研究程序的语义性质、语用性质、程序的开发与维护等,主要包括语义理论、语用理论、数据类型理论、程序逻辑理论、程序验证理论、并发程序理论,以及混成程序理论等。

此外,还有旨在借助计算机研究代数演算的“计算机代数”以及借助计算机研究数学定理证明的“计算机数学”等。

2. 计算机体系结构

计算机体系结构有两层含义。一为实体含义,二为学科含义。前者是指,从程序人员角度看到的计算机的组织结构与功能行为,组织结构指的是构成计算机的一级组成部分(如构成计算机的运算器、控制器、存储器、输入设备、输出设备)及其间的联系(意指这五部分如何协同工作,以完成各种计算);功能行为指的是计算机能进行且只能进行的所有基本运算所构成的集合,亦即,计算机的指令系统。亦有人认为,除上述者外,计算机体系结构的实体含义还包括计算机的实现(即其组织与硬件)。由于本书除计算机体系结构外,尚含有计算机硬件分支,二者不宜重复,因此,凡涉及物质实体者均在计算机硬件中讨论。后者是指在实体含义之计算机体系结构的研发过程中所涉及的理论、原则、方法、技术所构成的学科。

体系结构一词之英文是 architecture,该词源于建筑领域,其义为“建筑学”或“建筑物的设计和式样”。计算机体系结构(computer architecture)一词最早出现在 1959 年 L. R. Johnson 等人的著述中。1964 年 G. M. Amdahl 等人在介绍 IBM 360 系统时给出了“计算机体系结构”之定义,即为“程序设计人员所看到的计算机的属性,即概念性结构和功能特性”,或者亦可表述为上述之“从程序人员角度看到的计算机的组织结构与功能行为”。当时的“计算机”是单机。随着多机系统的出现,该定义的内涵亦有所发展。一个多机系统的组织结构是指构成该多机系统的所有单机所构成的集合及其各个元素之间的联系。多机系统的功能是其所属所有单机功能之并集。此后又有人将 G. M. Amdahl 等人给出的定义(实体定义)称为计算机系统体系结构(computer system architecture),而将计算机之实现(即组织与硬件)称为计算机实现体系结构(computer implementation architecture)。甚至还有人根本不同意将计算机的实现作为计算机体系结构的成分。

计算机体系结构主要包括:计算机、计算机组成、处理器结构、控制器结构、存储器结构、计算机可靠性、可用性、可维性(RAS)技术、性能评价、并行处理、高性能计算、网格计算、云计算、对等计算、移动计算、普适计算、分布式处理等。

3. 计算机软件

计算机软件的含义有三。一为个体含义,二为整体含义,三为学科含义。个体含义之计算机软件是指任一计算机系统中的任一程序及其有关的文档(程序是软件之主体,文档之作用是为了便于理解程序);整体含义之计算机软件是指任一计算机系统中所有个体含义之计算机软件所构成之集合;学科含义之计算机软件是指以整体含义之计算机软件为研究对象的学科,或者也可以说学科含义之计算机软件是指在实施个体含义之计算机软件过程中所涉及的理论、原则、方法、技术所构成的学科。

计算机软件之发展迄今可分为三个阶段。从世界上第一台计算机之第一个可用之程序出现开始至第一个可用之计算机高级语言出现为第一阶段(20 世纪 40 年代中期至 50 年代中期,确切地说,1946 年至 1956 年);自第一个可用之计算机高级语言的出现至软件工程的提出为第二阶段(1956 年至 1968 年);从软件工程提出迄今为第三阶段(1968 年迄今)。

计算机软件可分为系统软件、应用软件、支撑软件三类。系统软件是和应用领域无关的软件,如计算机操作系统,各种通用计算机语言的处理系统等。应用软件是和应用领域有关的软件,如购票软件、交通管制软件、人口普查软件等。支撑软件是支撑其他软件之开发与维护的软件,如软件支撑系统中之各种工具软件,中间件等。

计算机软件包括软件语言、软件方法学、软件工程、软件系统。软件语言是用以书写程序(或设计规约、功能规约、需求规约)和文档的语言,又可分为实现级语言(书写程序及其文档之语言)、设计级语言(书写设计规约及其文档之语言)、功能级语言(书写功能规约及其文档之语言),以及需求级语言(书写需求规约及其文档之语言)。软件方法学是以软件方法为研究对象的学科,着重研究诸如自顶向下方法、自底向上方法、面向对象方法、软件自动化方法、形式方法等。软件工程是应用计算机科学与数学原理开发与维护软件的工程,或者可以说软件工程的学科含义是用工程化方法研究、开发、维护软件的学

科，如软件开发环境、领域工程、需求工程以及软件自动化等。特别应该指出的是，迄今软件工程已成为软件中日益重要的标致性内容。软件系统包括各种用于软件开发与维护之系统以及应用软件系统，如操作系统、语言处理系统、数据库管理系统、计算机安全系统等。

4. 计算机硬件

计算机硬件之含义有三，一为个体含义，二为整体含义，三为学科含义。①个体含义之计算机硬件是指构成任一计算机系统的任一物质实体（如元件、器件、物件、设备等）。②整体含义之计算机硬件是指任一计算机中所含个体含义之计算机硬件所构成之集合。③学科含义之计算机硬件是指以个体含义的计算机硬件为研究对象的学科。

计算机之划代标志是元器件。第一代计算机是电子管计算机（20 世纪 40 年代中期至 50 年代末期）。运算器、控制器采用电子管，存储器采用阴射线管或超声延迟线、磁鼓、磁心，输入与输出设备沿用继电式计算机之穿孔卡片、穿孔纸带机和击打式打印机。第二代计算机是晶体管计算机（20 世纪 50 年代中、后期至 60 年代中期），运算器、控制器采用晶体管电路、存储器用磁心，输入与输出设备增加了磁带机和显示器，与第一代计算机相比，速度更快、容量更大、体积更小。第三代计算机为小、中规模集成电路计算机（20 世纪 60 年代中期至 70 年代初期），运算器、控制器采用小、中规模集成电路，一块芯片上只能集成 100 个以下的元器件，存储器采用磁心、大量使用磁盘存储器，输入与输出设备除沿用击打式打印机外，出现非击打式打印设备、磁带机，开始使用软盘，和第二代计算机相比，速度更快，容量更大，体积更小。第四代计算机为大规模集成电路（LSI）、超大规模集成电路（VLSI）计算机（20 世纪 70 年代中期以来）一块芯片上所能集成的晶体管可达 26 亿个，存储器也采用了 VLSI。

计算机硬件主要包括芯片、运算器、控制器、存储设备、网络设备、输入输出设备、工程设计与制造、硬件可靠性、硬件维护、机房设施等。

5. 计算机网络

计算机网络是地理上分散的多台自主计算机互连的集合。自主性排除了网络系统中各台计算机之间的主从关系。互连却需要遵循约定的通信协议。计算机网络可实现信息交互、资源共享、协同工作、在线处理等功能。此为计算机网络的实体含义。计算机网络的学科含义是指以实体含义的计算机网络为研究对象的学科。

自 1969 年美国国防部的国防高级研究计划署（DARPA）建立起世界第一个分组交换网 ARPARNET（即 Internet 之前身）以来，计算机网络之发展极为迅速，对人类之生产与生活起到了革命性的变革作用。一言以蔽之，计算机网络人人不可须臾离也。

计算机网络分类方式繁多，主要有如下几种。

按地域范围分，有局域网、城域网、广域网；按拓扑结构分，有总线网、星状网、环状网、网状网；按交换方式分，有电路交换网、分组交换网等；按网络协议分，有 TCP/IP、SNA、SPX/IPX、Apple TALK 等（其中以 TCP/IP 最为重要，IP 协议推动了各种应用，如 IP 电话等）；按安全控制政策分，有内联网、外联网等。

计算机网络主要包括网络体系结构、网络互联、数据通信、局域网、承载网、网络管理、网络安全、泛在网等。网络体系结构是构成网络的各台计算机之间相互通信的层次以及各层次中的协议和层次之间接口的集合。网络互联将多个网络互相连接,以实现在更大范围内的信息交换、资源共享和协同工作。数据通信是依据指定的规程或协议,将数据源的数据编辑后发送到目的地的过程和技术。局域网是将位于局部区域内的多台计算机互连起来的数据通信网络。承载网是由通信公司提供的一种网络,以便于形成联接若干城市、地区,甚至跨国,以至遍及全球的网络。网络管理包括对网络的配置、故障、性能、安全、计费等进行管理的功能。网络安全是为了发现并克服源于网络内部的漏洞与错误以及源于网络外部的安全威胁所采取的措施和技术。泛在网可理解成是一种可在任何时间、任何地点、为任何人及任何物提供顺畅通信联系的网络。

6. 计算机应用

计算机应用的实体含义是指计算机在各个领域的应用中所涉及的基本原理、有效方法与技术。计算机应用的学科含义是指以实体含义的计算机应用为研究对象的学科。

计算机所处理之信息包括数值信息与非数值信息,涉及到人类生产与生活的各个方面。早期的应用主要是科学计算与工程计算,稍后出现了狭义的数据处理问题,再后出现了非数值计算。

计算机应用主要包括中文信息处理、计算机图形学,数字图象处理、计算机辅助技术、多媒体计算技术、计算机控制、计算机信息系统、计算机仿真等。中文信息处理研究用计算机处理中文信息所涉及的原理、方法和技术。计算机图形学是借助计算机产生真实或虚拟物体图形的原理、方法和技术。数字图象处理是利用计算机将模糊或受损之图象进行处理,以实现图象之增强、复原、重建,以及分割、配色等的过程和技术。计算机辅助技术包括辅助设计、辅助制造、辅助工程、辅助教学等。多媒体计算技术包括用计算机综合处理文字、图形、图象、声音等多种形式之媒体信息。计算机控制是计算机用于实验、生产或类似过程中进行操作控制的过程和技术。计算机信息系统于此特指对信息进行采集、表示、储存、处理并以人机交互方式为用户提供信息服务的系统。计算机仿真是对各种类型的系统,根据它们的有关概念、变量、规则、逻辑关系、数学表达式、图形和表格等必要信息,建立数学模型或描述模型并在计算机上加以体现和试验,从而达到分析、研究该系统的目的的过程和技术。

建造计算机的目的在于用,在于在人类认识自然、改造自然、认识社会、改造社会中起积极作用。可以说,计算机硬件、计算机软件、计算机应用,三者的关系是,计算机硬件是物质基础,计算机软件是灵魂,计算机应用既是目的,又是发展之源泉,而计算机应用技术则是促进、推动计算机应用的重要力量。

7. 人工智能

人工智能有两层含义,一为实体含义,一为学科含义,实体含义的人工智能是指人类智能的模拟学科。智能是洞察、学习、思考、理解,以及推理的能力。学科含义的人工智能是指以实体含义的人工智能为研究对象的学科。

人工智能的任务有二,一是发展具有类似生物(人类)智能的计算机系统(即智能信息处理系统),二是借助计算机模拟生物(人类)的智能(认知科学)。

人工智能的主要研究内容包括知识表示、自动推理和搜索、机器学习和知识获取、知识处理、自然语言理解、计算机视觉、智能机器人、程序设计自动化等。表示是描述实际问题的框架,知识表示则是在表示的框架下对具体问题为真(即知识)的描述,这里的问题是,何谓知识?如本文基本概念一节中所述,迄今知识之形式定义有人刻画为一个三元组(事实集、规则集、置信度)。有专家知识之表示与常识知识之表示两种。推理是知识的使用过程,有演绎推理和非演绎推理两种。搜索是人工智能的求解方式,有盲目搜索和启发性搜索。机器学习是指在特定表示下借助计算机系统进行学习,以产生系统中原先不存在的知识或提高系统之解题能力,有归纳学习、统计学习等。自然语言理解包括对语法、语义、语用的理解,它是自然语言处理的核心。智能机器人是跨学科的领域,它涉及规划、推理、学习等方面。迄今已出现具有一定智能的机器人,在人类生产与生活中起到一定的辅助作用。程序设计自动化意指借助计算机系统使设计程序过程中之人工劳动尽可能减少,20 世纪 80 年代以来,国内外已出现一些实验性的程序设计自动化系统。

总之,20 世纪 50 年代以来,人工智能已经有了相当的发展。一方面,人们认识到这门学科的重要性,另一方面,其发展之道路相当曲折,虽有不少成果,但和“制造具有智能的机器”这一目标却相距尚远,但从计算机应用的角度来看,却又是成果甚丰,它已成为计算机应用发展的原始动力之一。

计 算 机 产 业

计算机产业的含义已在本文第一部分(即基本概念)中阐明,它是以计算机为产品的产业,其中包括计算机制造业、计算机服务业以及计算机软件产业三部分。计算机制造业从事计算机系统的生产制造。属于计算机制造业的企业有各种系统制造厂,外围设备和终端设备制造厂,记录媒体制造厂以及提供专用应用系统的厂家等。计算机服务业是为满足使用计算机的需要而提供服务的行业,它一般包括处理服务、专业服务和系统集成等方面,也包括计算机和有关设备的租赁、修理和维护等。计算机软件产业是从事计算机软件的开发、生产、销售以及售后服务的产业,它兼有制造业与服务业的双重特性。各种类型的计算机软件及相应机构都属于计算机软件产业。计算机产业可为国民经济和社会带来巨大的经济效益和社会效益,其发展水平和产业规模已成为衡量一个国家实力的重要标志。

世界计算机产业的发展可分为如下四个阶段。从 1946 年制成的 ENIAC 起至 20 世纪 60 年代中期为第一阶段。早期的计算机产品主要是由少数原来从事办公或商用机器制造的公司如 RemingtonRand、IBM 等开发生产而成,当时的计算机产品有 UNIVAC I,IBM 650 等。50 年代后期至 60 年代前期,美、英、日等国的计算机公司相继创立(如 CDC、DEC、Ferranti、富士通、东芝等),计算机产业开始从美国扩展到欧洲和日本,产品主要为面向军用或科学计算的大型机,产量较小,尚未形成规模产业。从 60 年代中期至 70 年代末为第二阶段。以 IBM 360 通用机系列及 DEC 的 PDP 小型机系列为代表的大批量

生产使计算机产业进入发展阶段。70 年代初期,以 Intel 为代表的微处理器芯片问世,并随之于 70 年代中期开发出微型计算机系统,进入市场,由此大大拓展了计算机的应用,促进了计算机产业的发展。同时,也使集成电路的研制生产企业成为计算机产业的重要组成部分。在计算机产业发展的相当长的一段时期内,计算机系统的硬件和软件一直是封闭式由同一企业配套开发的,二者密不可分。从 80 年代初期至 20 世纪末为第三阶段。IBM 采用 Intel 芯片与 Microsoft DOS 操作系统,开发出 PC 微型计算机系统,引起众多厂家进入兼容机的角逐。软件开发已不再与硬件开发紧密结合。80 年代开始,由于微型计算机、工作站和局域网技术的发展,更有力地推动了计算机产业的发展,使其地位迅速上升,成为各国经济发展中的重要的支柱产业。进入 90 年代,世界计算机产业进入了一个新的发展时期。软件产业异军突起,并以其技术发展牵动硬件技术的发展。由于精简指令集计算机等多种新技术的出现,世界计算机产业结构、产业布局和技术方向也发生了重大变化。技术发展一改过去专有封闭的做法,而走开放标准之路;产业也由过去由某一公司硬件、软件垂直开发的结构转变为硬、软件逐渐分离,并且多采用国际分工、专业化开发生产的结构。在 20 世纪的最后 10 年中,由于计算机网络大发展而导致计算机产业发展具有新特点:第一,微型计算机的需求量猛增;第二,多个领域(如电视、电影、电话、微机等)联合创建新产品,其特点是,体积更小、功能更强、使用更方便,如个人无线通信系统、便携式计算机、智能手机、智能信用卡等;第三,计算机产业在各国的国民经济中的比重猛增。至 20 世纪末,计算机产业已成为一种具有战略意义的产业。从 21 世纪开始迄今为第四阶段。步入 21 世纪后,由于技术的快速发展和市场的迅猛成长,计算机产业已经步入成熟期,计算机产品也成为人们生活的重要部分和基本工具。从 2000 年到 2008 年,全球 PC 出货量保持两位数增长,2009 年增量开始趋缓,2011 年增长率下降 1.6%。2012 年全球 PC 出货量为 3.487 亿台,迎来 11 年来首次下滑。2013 年全球 PC 出货量同比下滑 9.8%,只有 3.151 亿台。2013 年,平板计算机的兴起使得其出货量首次超过笔记本计算机,传统 PC 增长缓慢。2014 年全球 PC 出货量下滑趋势趋缓,全球平板计算机出货量预计为 2.5 亿台。

从处理器角度看,1965 年,Intel 创始人 Gordon E. Moore 提出“摩尔定律”,即“集成电路上可容纳的晶体管数大约每隔 18 个月便会增加一倍,性能也提高一倍”,50 年来这一定律一直维持有效,引领计算机产品性能呈指数级的提高。今后十年左右,“摩尔定律”还会继续发挥作用。但集成电路工艺换代速度会减慢,开始进入后摩尔时代。

我国计算机产业的发展亦可分为四个阶段。第一阶段(20 世纪 50 年代中期到 70 年代末期)的重点是根据国防建设与科学研究的需要,从借鉴苏制样机研究机制,逐步走向独立自主开发。科研成果即产品,为专门应用部门使用。因此,当时计算机生产厂家少、规模小、产品少而分散、发展缓慢。第二阶段(20 世纪 80 年代初期到 90 年代初期)国内进口了国外的微型计算机,在此基础上开发出 0300 系列、0500 系列等国产微型计算机。遵循引进、消化、开发、创新的方针,开发出一些小型计算机与工作站,以及 CC-DOS 汉字操作系统,发明了各种汉字输入与处理方法,汉化了 IBM、DEC 等公司生产的机器上使用的 VMS 和 DOS/VSE、MVS 等操作系统,开发了大量的应用软件和应用软件包。国内市场规模迅速扩大,计算机的年销售额由 1981 年的 5.2 亿元人民币增至 1990 年的 55.5 亿

元,计算机产业有了较大发展。第三阶段(20 世纪 90 年代中期至 21 世纪前几年)的显著变化是国际各大公司纷纷进入中国市场。国内企业向两极发展,一是扩大规模,继续发展自己的产品;一是向应用方向发展,针对用户需求开发应用系统。并自主开发 Office、Linux 等软件,进入市场。同时,中外合资企业及外资独资企业的大量出现,使外向型产业的规模迅速扩大,致使国内市场规模扩大更快,1996 年计算机产业的市场销售额为 920 亿元,2000 年为 2150 亿元,2003 年为 3327 亿元。中国计算机产业的总体水平迅速提高,规模迅速扩大,软件产业及信息服务业迅速发展,产业结构渐趋合理,基本形成了制造业、服务业和软件产业三大组成部分的格局。第四阶段(21 世纪初以来)我国计算机产业进一步发展壮大。2005 年我国计算机产业实现销售收入 10452.5 亿元,同比增长 21.7%;完成出口交货值 7989.3 亿元,同比增长 27.8%。此后,受世界经济发展放慢、国际金融危机影响,全球计算机市场增长减缓。2008 年,我国规模以上计算机行业实现销售收入 17134 亿元,同比增长 7.3%,规模继续在世界上保持前列,计算机产量约占全球产量的 40%。2011 年中国计算机行业规模以上企业销售收入为 21177.2 亿元,同比增长 16.14%。2012 年,我国规模以上计算机制造业产品产量达到 3.5 亿台,同比增长 10.5%,计算机产量超过全球出货量的 50%,稳居世界第一。值得注意的是,2012 年全球 PC 市场整体下滑,我国联想计算机公司的计算机生产却逆势增长,并取代国外巨头,成为世界 PC 行业排名第一的企业。2013 年,受海外市场需求不振、国内经济增长放缓等因素影响,我国计算机行业整体保持低速增长。由于智能手机的井喷式增长,对计算机行业冲击较大,市场竞争更加激烈,技术变革将改变原有市场格局,传统 PC 将向更便携、更具移动化趋势转变,产业结构面临深层次变革。2013 年,我国计算机行业实现销售产值 22401 亿元,同比增长 5.5%,2014 年,计算机行业实现销售产值 22729 亿元,同比增长 2.9%。

发 展 展 望

1. 计算机科学技术与通信科学技术紧密融合,相互渗透,大大加速人类社会信息化进程

随着世界各国信息基础设施的建立与发展,计算机科学技术与通信科学技术更加紧密融合,相互渗透,全球性的计算机联网促进信息资源的开发和利用。计算机进入千家万户,使它成为人类工作与生活的必需品。计算机科学技术成为人类必须学习的基础知识。特别是,网络计算、移动式计算、嵌入式计算、多媒体技术、虚拟现实技术、面向对象技术、主体技术等的有机结合与综合应用,展示出计算机与计算机科学技术的宏伟前景,必将出现计算机、计算机网络、计算机应用几乎处处可见的局面,从而大大加速人类社会信息化进程。

2. 新型元器件和体系结构的发展,以及相关技术的发展,大大提高计算机系统的性能,便于人类更好地认识自然

随着微电子加工技术的发展,半导体集成电路将会有更高的速度、更高的集成度和更高的性能价格比。计算机体系结构的发展将使计算机获得更高的性能。过去一些无法解

决的复杂问题将有可能解决，使人类能更好地探索自然和驾驭自然。

3. 新技术的研究、开发与利用，大大提高计算机软件的功能与性能，解决计算机系统开发中的软件瓶颈问题

随着以智能化、集成化、自动化、并行化、开放化以及自然化为标志的计算机软件新技术的深入研究、开发与利用，不仅使软件的功能与性能迅速提高，而且有可能从根本上解决软件生产率低下的问题。结合软件工程实践，探讨软件理论，有可能从理论上弄清软件开发的复杂度，进而采取有效措施进行控制，从理论与实践两方面来解决计算机系统开发中的软件瓶颈问题。

4. 新型计算机的研究与开发将受到更大的重视

由于集成电路之集成度不可能无限制地提高，它具有一定的上限，超出此上限，集成电路便无法稳定工作，从而，诸如量子计算机，生物计算机等新型计算机的研究与开发便日显重要，必将受到更大的重视。

5. 信息安全保密等成为计算机与计算机科学技术领域的重大课题

在全球联网的趋势下，为保证信息资源共享，计算机系统与网络的互操作性、开放性和标准化将受到高度重视。同时由于计算机进入千家万户，成为人人可以利用的设施，使用的简明化、自然化和信息安全保密等将成为计算机与计算机科学技术领域中的重大课题。

电子数字计算机是20世纪40年代人类的伟大创造。半个多世纪以来，计算机、计算机科学技术、计算机产业在世界范围内蓬勃发展，规模空前。它的诞生和发展对人类社会作用巨大，影响深远。今后宜继续本着造福于全人类的宗旨，遵循促进社会发展的方向，按世界各国相互学习、取长补短、互利互惠、共同提高的原则，阔步前进，为人类做出更大的贡献。

总论编写组
（徐家福执笔）
2015年9月

条 目 分 类 目 录

Ⅰ. 计算机科学理论

Ⅱ. 计算机体系结构

Ⅲ. 计算机软件

Ⅳ. 计算机硬件

V. 计算机网络

Ⅵ. 计算机应用技术

Ⅶ. 人工智能

A

Akeman hanshu

阿克曼函数（Ackermann's function） 一种递归而非原始递归的函数(参见**递归函数**和**原始递归函数**)。

阿克曼函数定义如下:

$$\begin{cases}\psi(0,y)=y+1\\ \psi(x+1,0)=\psi(x,1)\\ \psi(x+1,y+1)=\psi(x,\psi(x+1,y))\end{cases}$$

该函数定义中包含双重递归。当 $x>0$ 时,$\psi(x,y)$的值均可通过"较早"值 $\psi(x_1,y_1)$来定义。这里,或者 $x_1<x$ 或者 $x_1=x,y_1<y$。这一点可直观地说明该函数可计算。

ψ 是递归函数的严格证明参见文献[1],ψ 为非原始递归函数的严格证明参见文献[2]。

参考文献

1. Cutland N. Computability, an introduction to recursive function theory. Cambridge, UK: Cambridge University Press,1980

2. Peter R. Recursive functions. New York: Academic Press,1967 (陈火旺 贲可荣)

anquan caozuo xitong

安全操作系统(secure operating system) 具有强制访问控制机制并可验证的**操作系统**。强制访问控制机制是一种用于保障信息机密性、完整性和资源隔离性等特性的信息安全机制,与自主访问控制不同,强制访问控制是完全由操作系统而不是由信息的所有者(属主)决定的访问控制;可验证指可以通过某种分析或验证手段得出所实现的安全机制与所描述的安全策略是相符的。从广义上讲,凡是拥有较强抗安全攻击能力的操作系统都可以认定为安全操作系统。

沿 革

安全操作系统的研究始于20世纪60年代。1967年,计算机资源共享系统的安全控制问题引起了美国国防部的高度重视,隶属于国防科学部的计算机安全特别部队开始操作系统安全技术研究。1973年,出现了第一个可证明的信息安全系统的数学模型BLP以及支持BLP模型的安全操作系统Multics。1983年,美国国防部颁布了历史上第一个计算机安全评价标准,简称橙皮书。橙皮书对安全操作系统的研究与发展具有深远影响。90年代以来,随着网络技术的发展,操作系统的环境越来越复杂,1993年出现了更适合互联网环境的计算机安全评价标准Common Criteria,简称CC标准,1999年成为国际标准,并在美国和西欧相继推出了符合CC标准的一系列用于保护重要数据和应对安全攻击风险的安全操作系统。进入21世纪,基于开源的安全操作系统的研究活跃,出现了一批开源的安全操作系统,著名的有美国国家安全局为主开发的基于Linux的SELinux, Novel公司推出的AppArmor等。

操作系统安全等级分类

根据安全机制及安全保障强度的不同,操作系统通常可以分为不同的安全等级。按橙皮书的经典规范,操作系统一般可以分为D、C1、C2、B1、B2、B3和A1七个安全等级。每个等级对应确定的安全特性需求和保障需求,后一个安全等级涵盖前一个安全等级的安全机制,并提供更强的安全功能和安全保障。

D级是最小保护级。C1级是自主安全保护级,主要包含用户标识与鉴别,自主访问控制。C2级是受限的访问保护级,提供细粒度的自主访问控制和审计以及不允许遗留在内存或磁盘的信息(客体)被后面的进程或用户使用的"禁止客体重用"等。目前,人们广泛使用的商用操作系统多属C1级或C2级。

B1级是安全标记保护级,引入了访问的主客体标记和强制访问控制,并作为其主要特征。符合橙皮书B1安全级及以上评价标准的操作系统通常才称为安全操作系统,也称可信操作系统。B2级是结构化保护级,具有严格的安全策略及其形式化的模型,并提供可信路径以及对隐通道的分析。B3级是安全域级,设有专门负责安全总控的引用监控机,还拥有可信恢复机制。A1级是验证设计级,安全功能与B3相同,但形式化要求更高。

主要内容

安全操作系统包含的主要技术和安全机制通常有用户标识与鉴别、安全标记、强制访问控制、事件审计、禁止客体重用等,同时至少具有安全策略的非形式化描述,并对系统进行安全设计和验证。

用户标识与鉴别一般基于操作系统的可插拔的认证模块(PAM)实现。除了用户名与口令的认证机制外,进一步引入强化的身份认证,如基于智能卡的身份认证、基于指纹的身份认证等。安全标记是操作系统赋予访问主体(用户、进程等)和访问客体(如文件)安全属性的相应标签。用户标记由系统管理员通过操作系统设置,用户经认证后,获得用户标记,并派生进程标记;文件标记是赋予文件的安全属性,文件标记由创建文件的进程或者父目录决定。安全标记是强制访问控制的基础,为进行强制访问控制,还需要以规则形式规定主客体标记间访问关系的安全策略,强制访问控制一般采用策略实施与策略决策分离的原则实现,策略实施仅仅是嵌入到任务管理、文件系统、网络等代码中的一个控制程序,一旦进程需要访问客体,实施程序便通过调用策略决策模块进行权限的判断。基于安全策略逻辑的分析判断的实质性工作由策略决策模块完成,策略决策模块实际上是安全策略相关规则的一个通用解释程序。事件审计是对安全相关的事件进行审计,并可以在事后进行查看和分析。审计事件包括用户登录、文件访问、安全标记创建等。

发展趋势

随着信息安全与网络安全技术的发展,物理级、运行级、数据级和内容级等许多更有效的安全技术不断融入操作系统中。因此,只要能够为所管理的数据和资源提供相应安全级别的保护,并能有效控制硬件和软件功能的操作系统都可以认定为安全操作系统。典型的例子是 OpenBSD 操作系统,它不仅拥有很多专用的抗攻击功能,且每一行代码都经过严格细致的手工漏洞检查,被认为最安全的操作系统。可见,专门从安全的角度设计安全操作系统或者融入比强制访问控制更强和更丰富安全机制的操作系统将成为安全操作系统的发展趋势。近年来,为了适应网络应用的更高安全保障需求,广泛引入可信计算技术,从计算机体系结构入手,可信操作系统正成为构建基于信任根的计算系统信任链的重要研究方向。

参考文献

1. Wikipedia. http://en.wikipedia.org/wiki/Secure_operating_system

2. DoD 5200. 28-STD, Department of Defense Trusted Computer System Evaluation Criteria. National Computer Security Center, Ft. Meade, MD, USA, Dec 1985

3. 刘克龙,冯登国,石文昌. 安全操作系统原理与技术. 北京:科学出版社, 2004

(何连跃)

anquan shujuku

安全数据库(security database)　在通用数据库管理系统基础上,提高安全性标准的**数据库管理系统**。安全数据库涉及技术、管理和物理等多个层面,主要应对数据库系统的风险和威胁,保证数据机密性、一致性、可用性等特性。和通用数据库相比,安全数据库在数据备份、访问控制、数据加密、审计、推理控制等机制方面得到加强。

数据库系统的物理损坏来源于地震、火灾、洪灾、过热、闪电等因素。数据库系统主要通过数据备份策略提高其物理安全性。

数据库访问控制机制根据系统存储的访问控制规则,确保用户只能按照其授予的权限来访问和操作数据库中的信息。按照实现方式和保护强度的不同,数据库访问控制可以分为自主访问控制、强制访问控制和基于角色的访问控制等。自主访问控制机制基于用户-对象权限矩阵,比较用户的请求与矩阵中保存的权限,判定用户的请求是否合法。强制访问控制机制为访问数据的主体和数据库中的客体设置安全性标签,通过比较安全性标签的级别判定主体是否有权对客体进行进一步的访问操作,确保信息的单向流动只能由低安全级主体向高安全级主体流动。基于角色的访问控制通过引入角色来关联权限和用户,支持灵活的权限管理。

数据库安全审计系统提供了一种事后检查的安全机制。安全审计机制将特定用户或者特定对象相关的操作记录到系统审计表中,作为后续对操作的查询分析和追踪的依据。通过审计机制,可以约束用户可能的恶意操作。

数据库加密使用已有的密码技术和算法对数据库中存储的信息进行保护。加密后数据的安全性能够进一步提高。即使攻击者获取数据源文件,攻击者也很难获取原始数据。但是,数据库加密增加了查询处理的复杂性,由于可能采用不同的加密方法,同时加密后的数据很难保证原始数据的特性,数据库管理系统一般无法直接查询加密数据,查询效率

受到较大影响。加密数据的密钥的管理和数据加密对应用程序的影响也是数据库加密过程中需要考虑的问题。

数据库推理控制机制用来避免用户利用其能够访问的数据推知更高密级的数据,即用户利用其被允许的多次查询的结果,结合相关的领域背景知识以及数据之间的约束,推导出其不能访问的数据。推理控制目前尚处于理论研究阶段。

实现安全数据库管理系统不仅和数据库系统本身的安全机制相关,还和数据库系统所在环境和安全管理密切相关。数据库系统作为基于操作系统的一种系统软件,依赖操作系统提供的安全功能和机制。同时数据库系统和客户端的信息交换都通过网络实现。因此安全的操作系统、安全的网络、完善的安全管理机制是实现安全数据库管理系统的基础和前提。

参考文献

1. Silberschatz A, Korth Henry F, Sudarshan S. 数据库系统概念. 6 版. 杨冬青,李红燕,唐世渭,等译. 北京:机械工业出版社,2012

2. Liu L, Ozsu M T. Encyclopedia of database systems. Springer, 2009

(杨冬青　高军)

B

Bakesi fanshi

巴克斯范式（Backus normal form，BNF）
用以精确描述程序设计语言语法的一种形式体系。又称**巴克斯-诺尔形式**或**巴克斯-诺尔形式体系**。

美国 IBM 公司研究员 J. Backus 和丹麦哥本哈根大学教授 P. Naur 最早用以描述 ALGOL 60 语言的局部语法（其全程语法并不能用 BNF 描述），因而得名。

相对 BNF 所描述语法之语言说来，BNF 为**元语言**，它所描述语法之语言为对象语言。

BNF 之基本成分为如下部分。

元符号，一个是元等价符“∷ =”，表示“定义为”；一个是元或符“|”，表示对所指出的各款项之选择；还有一对尖括号“〈 〉”，用以连同所括之名表示非终极符。

元变量，用以表示相应对象语言之语法单位。元变量本身不属对象语言，其值为对象语言中之符号串。

以上述元符号及元变量为基础所构成的一组元公式（或称产生规则），用以刻画相应对象语言之局部语法。其形为

非终极符∷ =非终极符与终极符组成之符号串

其中非终极符（即上述之元变量）刻画对象语言之语法单位，终极符为对象语言之符号串。

元公式中“∷ =”之左方称为公式之左部，“∷ =”之右方称为公式之右部，具有相同左部之不同公式可以公用同一个左部。

例如，在 ALGOL 60 中，有

〈二进制数字〉∷ =0|1

〈十进制数字〉∷ =0|1|2|3|4|5|6|7|8|9

〈数字〉∷ =〈二进制数字〉|〈十进制数字〉

元公式之右部亦可出现正在定义的左方非终极符，此时为递归定义。如

〈标识符〉∷ =〈字母〉|〈标识符〉〈字母〉|〈标识符〉〈数字〉

即，标识符定义为以字母起头的字母数字串。

BNF 能严格表示一类上下文无关语言之局部语法，自从在 ALGOL 60 语言文本中首先采用以来，已在不少计算机学科中得到广泛应用和推广。

BNF 一词有其演变过程，起初代表 Backus normal form 或 Backus-Naur form，后来人们认为，既然 BNF 为一形式体系，为一元语言，称之为 form，似觉欠妥，F 以代表 formalism 为佳，从而便出现 Backus-Naur formalism，这里 formalism 一词与 formal system 同义。

参考文献

1. Naur P, et al. Report on the algorithmic language ALGOL 60. CACM, 1960, 3(5): 299-314

2. 陈火旺，等. 程序设计语言编译原理. 北京：国防工业出版社，1984

（陈火旺　程虎　贲可荣）

Bakesi-Nuoer xingshi tixi

巴克斯-诺尔形式体系（Backus-Naur formalism，BNF） 参见巴克斯范式。

bangong xinxi xitong

办公信息系统（office information system）
为政府和企事业等部门各级办公人员提供办公信息服务的**应用软件系统**。办公信息系统的作用是：以提高办公效率和办公质量为宗旨，利用部门所涉及的内外部信息，优化办公流程，改善办公信息的流转与管理，实现办公过程的信息化。在 2000 年 11 月办公自动化国际学术研讨会上，专家建议将办公自动化（OA）更名为办公信息系统。由此，一般认为办公信息系统与原名办公自动化系统是同义的。

办公自动化一词早在 1936 年就出现了，意在运用打字机和电话设备帮助办公人员处理办公业务。随着现代计算机的诞生和发展，特别是网络通信技术的发展，其外延和内涵都发生了很大变化。现代意义上的办公信息系统经历了 3 个阶段：①以简单数据处理为中心的阶段，表现为利用个人计算机和办公套件，实现数据统计和办公文档写作电子化，使办公信息载体从纸介质方式向电子方式转移。②以网络为基础、以工作流为核心的阶段，出现了文档管理、电子邮件、目录服务、公文流转和群组协同工作

等办公信息系统。③以决策管理为核心的阶段,不仅出现了网上实时交流信息的集成平台,实现公文和文档的一体化处理,而且在办公处理的每一环节上提供相关的决策知识,使办公人员从被动向主动转变,在提升办公人员决策能力的过程中,提高部门的整体办公质量、效率和应变能力。

办公信息系统由**计算机硬件**、**计算机网络**和办公专用设备,**操作系统**、**数据库系统**、**分布计算中间件**和办公套件等基础软件以及具体实现办公信息服务的应用软件组成。应用软件的开发参见**软件工程**。

办公信息系统的基本功能包括:①公文处理:实现公文的签收、登记、分发、归档等处理;②电子邮件:依据工作流程将文档传递给下一部门或人员;③签报管理:对公文进行拟稿、审核、签收、会签、拟办、签批、电子签名、交办、退稿、备查、销毁等管理;④会议与会务管理:提供会议计划、日程安排、会议资料准备、会议记录、会议查询和其他会务管理;⑤资料管理:对原始电子资料进行编辑、文摘与提要编写、刊号管理以及查询、统计和汇编;⑥案卷管理:将相关文件编辑成卷,并进行查阅和借阅管理;⑦公共信息服务:提供机构设置、职能与人员编制信息,各类业务信息以及通讯录、公告板等信息服务。此外,还有向公众提供信息发布、检索、查询、引导和专题电子论坛等功能。

在基本功能的基础上,可以按应用层次将办公信息系统分类为:①事务型,指日常办公事务处理和公文流转等行政事务处理两种;②信息管理型,指基于**管理信息系统**功能的办公信息系统;③决策支持型,指基于**决策支持系统**功能的办公信息系统。此外,还可以按应用行业特定的业务需求或特点分类为事务型、专业型、生产型、经营型、案例型和政府型等。

办公信息系统涉及软件技术、网络通信技术、行为科学和系统科学的综合应用。当前,办公信息系统已成为**电子政务系统**和**电子商务系统**的基础,在政府、企业和其他公共机构将得到更广泛更深入的应用。 (孙志挥 吴泉源)

bandaoti cunchuqi

半导体存储器(semiconductor memory) 用半导体大规模集成存储器芯片作为存储介质并能对数字信息进行存取的存储设备。典型半导体存储器的基本组成如图1所示,各部分的功能简述如下。

(1)半导体存储器芯片阵列 它是信息存储单元的集合,由**半导体存储器芯片**按一定结构组成,是供**中央处理器**(CPU)或输入输出设备(I/O)存取信息的存储空间。

(2)地址输入缓冲及驱动 由地址输入寄存器、片选译码器、地址分时电路和地址驱动器等几部分组成。地址输入寄存器的功能是接收CPU给出的地址,再将低位地址直接送给存储器芯片阵列,高位地址经过译码电路产生片选或块选信号。刷新计数器和地址分时电路为动态随机存取存储器(DRAM)所必需,它们的功能是将器件输入的行地址和列地址归并为一路,并将所产生的刷新地址送至存储器芯片的地址端口。

(3)数据锁存及双向驱动 包含输入缓冲门、输入数据寄存器、输出锁存器和输出缓冲门等。它的功能是根据CPU的读写命令,将数据总线的内容写入被访问的存储单元,或者将被访问存储单元的内容读出到数据总线上,供CPU或I/O使用。

(4)控制电路 包括访问信号控制和刷新电路。它的功能是接收CPU的启动、读、写、清除等命令,并对接收后的命令进行处理,产生各种时序控制脉冲和内部定时脉冲信号来控制存储器的读写操作或刷新操作。

在半导体存储器的组成中,存储器芯片阵列是核心,其余三部分是存储器的外围电路。

半导体存储器按在计算机中的工作性质可分为

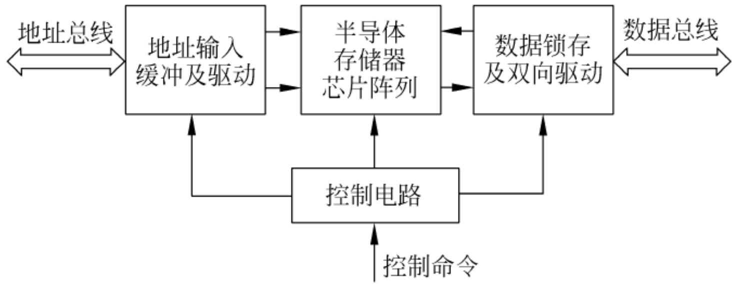


图1 典型半导体存储器的基本组成框图

主存储器(内存)、**辅助存储器**(外存)、**高速缓冲存储器**和控制存储器等。主存储器具有速度高、存储容量较小的特点,用来存放 CPU 当前执行的程序和数据。通常主存储器是由随机存取方式和只读方式两种半导体存储器组成,它们分享主存储器的同一地址空间。辅助存储器的特点是速度慢,存储容量大,用来存放 CPU 暂时不用的程序和数据。磁盘是辅助存储器的主流,但半导体盘(**固态盘**)已能实现磁盘的功能,并已在一些应用领域取代了磁盘。从功能上看半导体盘也可由电荷耦合器件(CCD)存储器实现,但从容量、速度和成本等因素综合考虑,则宜采用由 MOS 动态存储器(DRAM)或快可擦编程只读存储器构成。缓冲存储器用于在两个不同工作速度的部件之间交换信息过程中起缓冲作用。高速缓冲存储器是一种容量较小、速度很高的存储器,是为了克服主存储器与 CPU 在数据传输速度上的不匹配而设置的,通常采用双极型半导体存储器芯片来制作。磁盘缓冲存储器是为克服主存储器与磁盘存储器之间数据传输速度的不匹配而设置,利用磁盘缓存的方法可缩短磁盘平均取数时间,进而提高磁盘系统使用效率。磁盘缓冲存储器通常用 MOS DRAM 构成。

半导体存储器种类很多,就其制造工艺而言可分为**双极型半导体存储器**和**金属-氧化物-半导体存储器**(简称 MOS 存储器)。双极型半导体存储器以双极型触发器作为存储单元,其中采用晶体管-晶体管逻辑存储单元的称为 TTL 型存储器,采用射极耦合逻辑存储单元的称为 ECL 型存储器。MOS 存储器以金属-氧化物-半导体场效应晶体管作为存储单元。MOS 存储器的特点是集成度高,工作速度较快,适用于作容量较大的主存储器。电荷耦合器件(CCD)也是金属-氧化物-半导体存储器件,基于此器件构成的存储器称为电荷耦合器件存储器。它是一种易失性串行存储器,通常为非破坏读出,仅当写入新信息时,才清除原来存储的信息。

半导体存储器按存取方式可分为随机存取存储器(RAM)和**只读存储器**(ROM)两大类。

1. 随机存取存储器　RAM 可以随机地按指定地址从存储单元存取数据,在半导体存储器出现以前,主要是以记忆磁心为存储单元的磁心存储器。1971 年美国 Intel 公司推出 1103 型 1 kb MOS DRAM 芯片后,半导体存储器开始在一些主要厂家用作计算机的主存储器,到 1976 年 MOS RAM 主存储器的每信息位的价格已降到磁心存储器的一半,从此 MOS RAM 取代了磁心存储器,并一直占据主导地位。RAM 还可进一步分为**静态随机存取存储器** SRAM 和**动态随机存取存储器** DRAM 两类。

静态随机存取存储器　一种使用双稳态锁存电路存储每个信息位的半导体随机存储器。它无须像动态存储器一样周期性刷新信息位,但当没有供电情况下,其保持电荷仍然会丢失,因此在一般意义上它也还是易失性的(参见**静态随机存取存储器芯片**)。SRAM 主要用于二级高速缓存(level2 cache),或者处理器和主板存储器之间的缓存。SRAM 也有许多种,如异步 SRAM、同步 SRAM、流水式突发 SRAM (pipelined burst SRAM)等。

动态随机存取存储器　利用存储单元的电容内存储电荷的多寡来代表一个二进制比特(bit)是 1 还是 0 的半导体存储芯片构成的存储器。由于在现实中电容会有漏电的现象,导致电位差不足而使记忆消失,因此除非电容经常周期性地充电,否则无法确保记忆长存。由于这种需要定时刷新的特性,因此被称为"动态"存储器。DRAM 主要用于作计算机的主存储器。

基本 DRAM 存储单元结构(参见**动态随机存取存储器芯片**)多年来都没有太多的改变,但是 DRAM 间通信接口一直在变化。包括 asynchronous DRAM (异步 DRAM),视频 DRAM(VRAM),窗口 DRAM (WRAM),快页模式 DRAM(FPM DRAM),扩展数据输出 DRAM(EDO DRAM),突发 DEO DRAM(BEDO DRAM),直接 rambus DRAM (DRDRAM),同步 DRAM(SDRAM),双倍速率同步 DRAM,四倍速率同步 DRAM(DDR 2),八倍速率同步 DRAM(DDR 3)等。

2. 只读存储器　在正常运行中只能随机读取预先存入的信息而不能写入新的内容,也即信息一旦写入就不能更改。即使在断电情况下,ROM 仍能长期保存信息内容不变,所以它是一种永久存储器。

随着大规模集成电路集成技术的发展,出现了多种大规模集成电路 ROM 芯片,其中掩膜只读存储器(mask ROM)结构简单,存储信息稳定,可靠性高,能够永久性保存信息。可编程只读存储器(PROM)是由半导体厂家制作"空白"存储阵列(即所有存储单元全部为 1,或全部为 0 状态)出售,用户根据需要可以实现现场编程写入,但只能实现一次编程。可擦编程只读存储器(EPROM)、电可擦编程只读存储器(EEPROM)和快闪可擦编程只读存储器(flash EPROM)等 ROM 不仅可以现场编程,还可

以擦除原存储的信息内容,写入新的信息。目前使用的 ROM 都是大规模集成电路存储器芯片,当它们装入程序或微程序后就称为固件。

半导体存储器自 1971 年以来发展迅猛。以 MOS RAM 为例,集成度平均以每三年增加 4 倍的速度增长,存取周期缩短,逻辑功能加强。在 MOS RAM 中,DRAM 的集成度一直约为 SRAM 的 4 倍,因此 DRAM 多用于大容量存储器中。由于 DRAM 存储单元利用电容存储电荷的机理保存信息,故使用 DRAM 时必须定时周期性刷新所存储的信息,以免丢失。SRAM 不需要刷新操作,使用简便,在一些容量稍小的存储器中使用较广泛。由于 MOS SRAM 的速度不断提高,已几乎替代了双极型 RAM,促使双极型 TTL RAM 向更高速度方向发展。

半导体存储器由于存储单元阵列及其外围电路可集成在同一芯片上,因而生产过程简单,调试方便,容易实现自动化。它具有可靠性高,结构紧凑,组装密度高,体积小,工作速度快等特点;其缺点是信息易失性,所存信息因断电而消失。在不允许信息消失的应用场合,必须采取断电保护措施。但半导体存储器的综合优势是其他类型存储器不能相比的,在相当长时期内其性能与应用范围还将有较大发展。

参考文献

郑筠. MOS 存储系统与技术. 北京:科学出版社,1990
（郭志先　曹强）

bandaoti cunchuqi xinpian

半导体存储器芯片(semiconductor memory chip) 用半导体集成电路制造工艺把很多存储单元电路排列成阵列,并和外围电路集成在同一硅晶片上,形成能储存大量数据的集成电路。它广泛用于计算机、通信和家用数字电器等数字系统。

在数字系统中,指令和数据(包括声音、图像等信息)都以二进制数的形式出现。每个存储单元通常存放一位二进制数。在半导体存储器(SM)芯片中,通常把存储单元排成由行/列寻址的存储单元矩阵(阵列),存放数字系统工作过程中需暂时或长期保存的二进制数。为了存入和取出需要的数,存储阵列的外围电路包括选取存储单元的行/列地址缓冲和译码驱动电路、读出数据放大电路、写入数据驱动电路、片选和内部时序控制电路等。存储器芯片上集成了存储单元阵列及这些外围电路。这种行列结构的单元阵列给 SM 芯片带来一个共同的特点,当输入的行地址译码后,只激励一条选中的行线,其他行均为未选状态。与选中行相连的所有存储单元同时将所存的数据输出到相应的列数据缓冲寄存器,再由已输入列地址缓冲的列地址译码后将选中列的数据输出到相应的数据输出引脚;若再读出同一行地址而列地址不同的存储单元数据时,只需输入列地址,或由芯片按设定规则改变列地址,就可快速输出相应的数据。而写入时,在行地址输入后,可将仅列地址不同的数据串,根据列地址的变化写入对应的列数据缓冲中,其他数据缓冲保留行选中时读出的数据;再将数据缓冲中的数写入到选中行对应的单元中。这就是 SM 芯片发展过程中,为提高数据传送速率而采用的页面方式、同步方式的成组传送和双倍数据速率传输等技术的基础。

由于半导体存储器(SM)芯片由大量相同的存储单元组成,电路逻辑结构简单,特别易于大规模集成,成为半导体大规模集成电路产品市场非常重要且不可缺少的部分。在计算机,手持通信和娱乐设备,家用电器等数字系统中得到越来越广泛的应用。不仅有单独的 SM 芯片,而且在包括 CPU 在内的系统集成的芯片上也集成了不同类型的存储器电路。从 20 世纪 60 年代末以来,随着半导体工艺的发展,SM 芯片也经历了由双极型、P/NMOS(P/N 金属氧化物-半导体)型到互补金属氧化物-半导体(CMOS)型 SM 的发展历程。CMOS 工艺由于极低的静态功耗和大规模集成的方便,成为现今 SM 芯片的主流。

基本 SM 芯片分类如图 1 所示。根据断电后对数据的保存能力,SM 芯片可分为**易失性存储器芯片**和**非易失性存储器芯片**两大类。易失性存储器又分为随机存取存储器与非随机存取存储器。非随机存取存储器多以模块方式集成在芯片中,较少以单独芯片方式出现。而且其中的先进先出(FIFO)存储器、后进先出(LIFO)存储器的内核是静态随机存取存储器(RAM),用作高速缓存的关联存储器,也称按内容寻址存储器(CAM),其存储单元也是 SRAM 单元。所以易失性存储器芯片一般指随机存取存储器(RAM)芯片(参见**随机存取存储器芯片**)。它可从任何给定地址的存储单元快速读出或写入数据,但断电后数据消失,再加电后数据不确定。非易失性存储器芯片在断电后所存储的数据仍能保存,再加电后所存数据仍可继续使用,通常指只读存储器(ROM)芯片(参见**只读存储器芯片**)。ROM 在初期通常存储引导程序、基本操作系统和常用的不变的数据。因而将 ROM 数据的输入称为编程。

- 半导体存储器芯片
  - 易失性存储器芯片（掉电后数据消失）
    - 静态随机访问存储器（SRAM）芯片
      - 非同步SRAM芯片（通常的SRAM芯片）
      - 同步SRAM芯片
        - 单倍速数据速率（SDR）芯片（SSRAM芯片）
        - 二倍速数据速率（DDR）芯片（根据时钟频率范围的不同有DDR，DDR2，DDR2+，芯片）
        - 四倍速数据速率（QDR）芯片（根据时钟频率范围的不同有QDR，QDR2，QDR2+，芯片）
    - 动态随机访问存储器（DRAM）芯片
      - 非同步DRAM芯片（通常的DRAM芯片）
      - 同步DRAM芯片：单倍速数据速率（SDR）芯片（SDRAM芯片）
        二倍速数据速率（DDR）芯片（根据时钟频率范围的不同有DDR，DDR2，DDR3芯片）
  - 非易失性存储器芯片（掉电后数据保存）
    - 不可擦除只读存储器（ROM）芯片：掩膜型（Mask）ROM芯片，熔丝型一次编程ROM（PROM）芯片
    - 可擦除只读存储器
      - 脱机编程：紫外光擦除可编程ROM（UV-EPROM）简称EPROM
      - 在线编程或电可擦除可编程ROM（EEPROM）
        - 串行EEPROM芯片：I^2C、SPI和Microwire总线的芯片
        - 快闪EPROM（Flash EEPROM）芯片
          - NOR Flash（并行EEPROM）
          - NAND Flash 快闪存储器

图 1　基本半导体存储器芯片分类

RAM 芯片分为静态 RAM（SRAM）芯片（参见**静态随机存取存储器芯片**）和动态 RAM（DRAM）芯片（参见**动态随机存取存储器芯片**）。SRAM 具有简单的接口、快速的地址读出时间和读出写入周期，这些优点使其作为高速缓冲存储器的应用一直延续至今。但 SRAM 偏大的存储单元面积和功耗的缺点，使 SRAM 芯片的存储容量比不上后来出现的 DRAM 芯片。DRAM 采用单管存储单元，数据存储在与单管相连的电容中，在读出后数据会被破坏，需重新写入。长时间不读时数据也会因漏电而消失，必须周期性地对所存数据进行刷新（读出/重写），以防止数据消失。这就是称为动态 RAM 的原因。而 SRAM 芯片则不需要刷新。

随着 CPU 工作频率的不断提高，对高速缓存和内存速度和容量的要求也不断提升。除工艺技术的进步外，SRAM 和 DRAM 都先后增加时钟输入，以同步地址、命令和数据接口。出现了同步 SRAM（SSRAM）和同步 DRAM（SDRAM）芯片，而内核仍是非同步的 SRAM 和 DRAM 电路。再进一步提高数据传输速率的技术是双倍数据速率（DDR），即相对于输入时钟，其上升和下降边沿都是接收数据的时钟。这样，在突发传送时，数据的传送速率是芯片输入时钟频率的二倍。此外，四倍数据速率（QDR）SSRAM 将数据输入端口和数据输出端口分开，利用流水线操作交替输入读出命令及地址和写入命令及地址。读出数据和写入数据可分别以双倍数据速率从各自的端口输入和输出。这样，芯片的最大数据输出速率可达时钟频率的四倍。

非易失性存储器中的只读存储器（ROM）芯片分为不可擦除 ROM 芯片和可擦除 ROM 芯片两种。掩膜型 ROM 芯片和熔丝型可编程 ROM（PROM）芯片（参见**可编程只读存储器芯片**）是最初使用的不可擦除 ROM 芯片。最先应用的可擦除可编程 ROM（EPROM）芯片是紫外光可擦编程 ROM（UV-EPROM）芯片，简称 EPROM 芯片。EPROM 芯片用紫外线擦除整个芯片所存数据，然后重新写入所需的数据。之后出现的电可擦除可编程 ROM（EEPROM）芯片（参见**电可擦编程只读存储器芯片**）用高电压实现擦除和写入，可实现在线擦除和编程。后来开发的技术可实现存储块或全芯片擦除，这就是快闪（快擦除）EPROM（Flash EPROM）芯片（参见**快可擦编程只读存储器芯片**），简称快闪存储器或闪存芯片。根据接口和内部组织的不同，闪存芯片又分为 NOR 和 NAND 两种类型。NOR 闪存芯片读出速度快，但存储单元面积大，不适合大容量集成。而 NAND 闪存芯片读出速度慢，但存储单元面积小，适合大容量集成。

与 RAM 不同，EEPROM（包括 NOR 和 NAND 闪存）的擦除/写入次数是有限制的，一般为 10^5 或 10^6 次。写入数据的保存时间可达 10 年。而且写入前要先擦除，写入时间和擦除时间都偏长，特别是擦除时间更长。因而不能作普通 RAM 使用。

为了满足客户/市场的需要，将满足客户特殊需要的外围电路与所需的核心 SM 电路集成在同一芯片上，形成了一些专门的存储器芯片。例如伪静态 RAM 芯片（或称为单管 RAM 或 Ut RAM 芯片）就是单管单元 DRAM 的内核和 SRAM 的接口，再加上 SRAM 与 DRAM 的接口转换和 DRAM 的时序控制，集成在同一芯片而成。目的在利用 SRAM 接口的方便性和 DRAM 易于大规模集成与成本低的优点。但随着技术的发展 Ut RAM 芯片现已很少单独应用。又如 OneNAND 芯片具有 NOR 闪存的接口和速度和 NAND 闪存的容量，就是在 NAND 闪存芯片上又嵌入集成了接口转换时序控制、小容量 SRAM 数据缓存和出错校验电路。可将原来存程序的 NOR 闪存芯片和存数据的 NAND 闪存芯片合为一特殊的 NAND 闪存芯片，用于要求体积小的手持/移动设备上。

用多芯片封装（MCP，参见**封装内系统**）也可形成一些专门的存储器芯片，以减小封装尺寸（例如 Ut RAM + NOR 闪存，NAND + DDR2 等），适应手持设备限制体积的要求或扩大存储容量，例如由 2/4/8 片 16 Gb/片封装成 32/64/128 Gb 的芯片，即多芯片的 DDP/QDP/ODP 方式，以满足大容量多媒体存储卡和固态硬盘的需求。

总的来说半导体存储器芯片向更大容量、更快速度、更多功能和更低功耗方向发展。为此随着工艺设计尺寸的不断缩小，MOS 管和存储单元的面积也不断减少。这为加大容量，提高速度，集成更多的功能创造了条件。同时允许的工作电压也不断降低（由初期的 5 V 降到现在的 3.3/2.5/1.8/1.5/1.35 V），功耗也可降低。为对这些基本的 SM 芯片的容量和时间参数与应用的关系有一个概念，用表 1 举例说明。

参考文献

1. Sharma A K. 先进半导体存储器——结构、设计与应用. 曾莹，等译. 北京：电子工业出版社，2005

2. 桑野雅彦. 存储器 IC 的应用技巧. 王庆，译. 北京：科学出版社，2006

3. http://www.samsung.com/Products/Semiconductor

（孙祖希）

表 1 基本半导体存储器芯片的参数和应用实例

SM 芯片名称	QDRII SSRAM	DDR3 SDRAM	NOR 闪存	NAND 闪存
典型容量	1 M×36 b	2 Gb	4 M×16 b	(256 M×16 b) + 16 MB ECC
时钟周期	4.0 ns	1.5 ns	读周期 70 ns min	读周期 42 ns min
行地址读出时间/随机读出周期	4.45 ns/8 ns	27 ns/48 ns	70 ns/70 ns	随机读 60 μs max
突发方式数据读出时间/读出周期	0.45 ns/2.0 ns	数据输出速率 1333 Mb/s	70 ns/70 ns	顺序访问 42 ns
突发方式写入时间	突发方式写命令周期 8 ns/4 字	数据输入速率 1333 Mb/s	字编程 14 μs 块擦除 0.7 s	页面编程 930 μs 块擦除 16 ms max
工作电源	1.8/1.5 V	1.5/1.35 V	5～1.8 V	3.3/1.8 V
主要应用	高速缓存	主内存	系统启动和基本操作系统	断电后需保存的大量数据和程序

ban jiandu xuexi

半监督学习（semi-supervised learning） 研究如何综合利用有标记样本和未标记样本的**机器学习**理论、模型与方法。机器学习算法都有输入和输出（例如分类问题输入就是数据特征，输出就是数据标签），其目的就是学习一个从输入到输出的映射函数。半监督学习研究如何在已知数据的输入和其中一部分数据的输出的情况下有效准确地学习映射函数。根据映射函数的类型，半监督学习又可分为直推（transduction）和归纳（induction），其中直推

设置下的半监督学习往往只能预测已知输入样本的输出,而不能学习具体的映射函数的形式,因而这类算法往往不具有拓展性。而归纳设置下的半监督学习能够得到映射函数的具体形式,因而具有更好的拓展性,但是具体算法也更加复杂。

传统的机器学习算法包含监督学习和非监督学习两大类。其中监督学习假设学习映射函数时已知所有输入数据的对应输出,而其目的就是学习一个具体的映射函数。非监督学习(参见**非监督学习**)假设仅仅已知输入数据,而没有任何输出的信息,其目的是从这些输入数据中学习有意义的输出。监督学习和非监督学习都有着各自的优势和局限性。监督学习由于有相应的输出监督信息,使得学习到的结果更加准确可靠,但是得到这些监督信息往往需要请相关领域专家标注,这就需要花费人力财力。相反,非监督学习不需要任何的监督信息,这也使得学习到的结果往往不可靠。而半监督学习则综合了以上两种学习算法,它往往只需要一小部分输入数据的输出,却能够得到与监督学习相当甚至更好的学习结果。

迄今为止,学者们已经提出了很多种半监督学习的算法,例如基于产生式模型的算法,低密度分离,以及基于图的算法等。半监督学习中也还有很多有争论的问题,例如没有监督信息的样本究竟什么时候会有帮助,以及算法的推广性能如何等。

参考文献

1. Chapelle O, Schölkopf B, Zien A. Semi-supervised learning. MIT Press. 2006

2. Zhu X, Goldberg A B. Introduction to semi-supervised learning. Synthesis Lectures on Artificial Intelligence and Machine Learning, 2009, 3(1): 1-130

(王飞　张长水)

ban jiegouhua shuju

半结构化数据(semistructured data)　介于完全结构化数据(如**关系数据库**与**面向对象数据库**中的数据)和完全无结构数据(如音像数据与文本数据)之间的数据。

和关系数据不同,半结构化数据一般是自描述的数据,它的模式不是统一的、预先给定的,而往往是不规则且经常变动的,并且它的数据和模式是混合存放的。这种特点使得半结构化数据有很大的灵活性,能够满足不同应用、不同企业之间交换信息的需要,但同时也给半结构化数据存储、查询处理带来了很大困难。目前得到广泛应用的 XML 数据就是半结构化数据。

对象交换模型(object exchange model, OEM)可以用来描述半结构化数据。它最初在斯坦福大学的异构数据集成系统 TSIMMIS 中引入。OEM 是一个简单的、自描述的、嵌套的对象模型。每个对象具有唯一的对象标识。对象分为原子对象和复合对象两种。原子对象包含一个原子类型的值,如整数、实数、字符串等。复合对象的值是有序对〈标记,子对象标识〉的集合,每一个标记是对象与子对象之间关系的描述。从结构上看,OEM 是一个带标记的有向图,节点表示对象,边表示对象间的关系。每个 OEM 图都有一个根节点,从根节点至任意节点都是可达的。

图 1 是对象交换模型的一个简单示例。它描述了某个餐饮业的结构。图中每个对象都有一个整型的对象标识,根对象包括三个子对象,两个餐馆和一个酒吧。每个餐馆都是复合对象,而酒吧是一个原子对象,字符串“酒吧 1”是该对象的值。

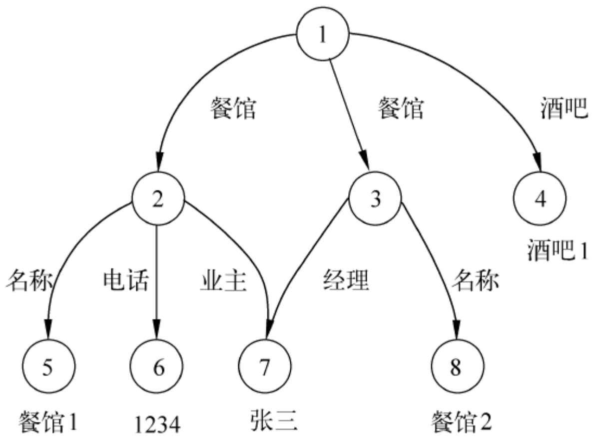


图 1　对象交换模型示例

半结构化数据管理中查询处理需要支持从现有的半结构化数据中定位特定部分,或者将一种类型的半结构化数据转换为另一种类型的半结构数据。为了提高半结构化数据查询的效率,通常需要建立面向数值或者面向结构关系的索引。

数据模式抽取和增强是半结构化数据管理中一个重要操作。为了有效地管理大量存在的无结构数据和半结构化数据,我们可以抽取或增强其模式信息,从而支持更加丰富的查询类型和精确的查询处理,实现数据优化存储,支持数据有效性验证等。

在半结构化数据存储过程中,一方面需要保持半结构化数据的结构信息,支持结构查询处理;另一

方面,需要充分考虑半结构化数据模式灵活的特点,支持半结构化数据结构模式的演化。

半结构化数据的一个主要应用是数据发布。由于半结构化数据模式比较灵活,不同来源的数据都可以转换为半结构化数据,从而在不同机构之间实现信息交换和共享。半结构化数据的另一重要应用是数据集成。随着半结构化数据的发展,很多领域都制定了相关的模式标准,可以基于这些领域模式实现异构数据的集成。

参考文献

1. Buneman P. Semistructured data. Arizona: ACM Press, 1997

2. Abiteboul S. Querying semi-structured data. Greece: Springer-Verlag, 1997

3. Liu L, Ozsu M T: Encyclopedia of database systems. Springer, 2009

（高军　杨冬青）

bangding

绑定(binding)　一个对象(或事物)与其某种属性建立某种联系的过程。如一个变量与其类型或值建立联系,一个进程与一个处理器建立联系等。这种联系的建立,实际上就是建立了某种"约束"。

绑定主要用于**程序设计语言**、**数据库**等方面。

在程序设计语言中,它指把数据名转换为机器地址的过程,把类型或值赋予变量或参数的过程等。

在**数据库系统**中,则指把数据的一种视图转换为另一种视图的过程。在应用程序可以使用数据之前,先要把应用程序的局部逻辑视图(子模式)绑定到数据库的全局逻辑视图(模式),尔后再绑定到物理视图(存储模式)。绑定可以早在编译时进行,也可以迟至取数据时进行。一般而言,及早进行绑定可提高运行效率,但适应修改的灵活性就较差;延迟绑定则可取得相反的效果。程序设计语言的设计必须在灵活性和效率之间求得平衡,绑定时间的控制正是达到这种平衡的一个重要手段。

绑定还分***静态绑定***和***动态绑定***,静态绑定只需检查程序正文就可以判定绑定出现与给定的应用出现是否对应。至于动态绑定,绑定出现与给定的应用出现是否相对应要取决于程序的动态控制流,LISP 和 Smalltalk 等语言有动态绑定,即有动态类型,大多数程序设计语言(包括 FORTRAN,ALGOL,Ada,PASCAL 等)均选择了静态绑定。

参考文献

岳东,李南. 微型计算机高级程序设计语言的分类与剖析. 北京: 海洋出版社,1992

（程虎）

beiyesi wang

贝叶斯网(Bayesian network,BN)　一个有向无环图,其结点表示随机变量,结点之间的边表示变量间的直接依赖关系。每个结点 X 都附有一个条件概率分布 $P(X|\pi(X))$,其中 $\pi(X)$ 表示 X 的父结点集合。是知识系统中处理不确定性的一种内涵方法(参见**不确定性推理**)。

1988 年,Judea Pearl 首次提出了贝叶斯网的概念并将其用于处理人工智能中的不确定性。由此,1999 年他获 IJCAI 终身成就奖。贝叶斯网是一种基于概率的不确定性推理网络,属于**概率图模型**(Probabilistic Graphical Model),它已成为不确定性处理的一种重要方法和人工智能的一个重要研究方向。近年来,贝叶斯网已广泛应用于**数据挖掘**、**机器学习**、图像处理、数据融合和决策等领域。

图 1 是一个贝叶斯网的例子。它是一个肺病诊断模型,其含义为:出访某地(V)可能感染肺结核(T);吸烟(S)可能引发肺癌(L)和支气管炎(B);肺结核(T)和肺癌(L)都可能导致肺部异常(C);肺部异常(C)和支气管炎(B)都可能引起呼吸困难(D);肺部异常(C)还可能会使 X 光检查结果呈阳性(X)。问题域中共包含 8 个随机变量 V、S、T、L、B、C、X、D,每个随机变量具有真、假(t、f)两种取值状态。

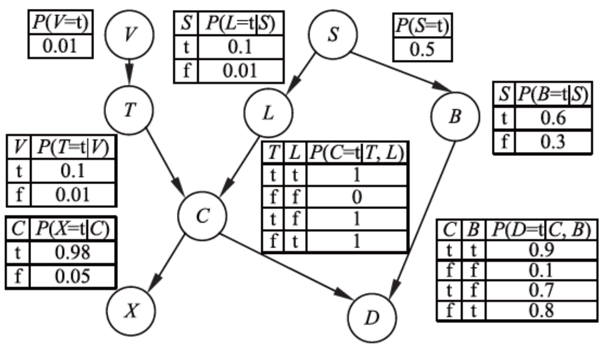


图 1　贝叶斯网结构及结点的概率分布图例

贝叶斯网研究主要包括推理和学习两方面。贝叶斯网推理系指通过贝叶斯网中的已知证据变量值,计算另外一些未知的查询变量的后验概率分布;贝叶斯网学习是指通过机器学习的方法从数据中获得贝叶斯网。贝叶斯网推理和学习都是 NP-难

解的。

贝叶斯网推理包括精确推理和近似推理。变量消元法和团树传播法(clique tree propagation)是应用较广的两种精确推理方法。通常,精确推理方法的计算复杂度很高,因此往往降低对精度的要求采用近似推理算法。随机抽样法、变分法、模型简化法和基于搜索的算法等是几种常用的近似推理算法。

贝叶斯网由网络结构和条件概率分布(亦称为贝叶斯网参数)组成,所以贝叶斯网学习包括参数学习和结构学习两部分。参数学习是指在已知网络结构的条件下,学习每个变量的条件概率分布,基本的方法主要有极大似然估计和贝叶斯估计。当数据不完备时,常采用 Gibbs 抽样方法、EM 算法和梯度方法等进行学习。结构学习是指学习贝叶斯网的拓扑结构,即有向无环图。主要有两类方法：基于评分的方法和基于约束的方法。基于评分的方法需要定义关于网络结构的评分标准,并选择合适的搜索算法在候选网络结构中搜索分值最高的网络结构。网络结构的评分需综合考虑与数据的匹配程度以及网络结构的复杂程度,如常用的最短描述长度(minimum description length,MDL)、贝叶斯信息准则(Bayesian information criterion,BIC)、AIC 信息准则(Akaike information criterion,AIC)等。基于约束的方法首先找出数据中蕴涵的各变量间的条件独立性关系,然后寻找与这些条件独立性关系一致的网络结构。

提高贝叶斯网的推理和学习效率、提高贝叶斯网表达能力以及进一步拓展其应用范围是贝叶斯网的主要发展趋势。

参考文献

1. Pearl J. Probabilistic reasoning in intelligence systems：Networks of plausible inference. Los Altos, CA：Morgan Kaufmann Publishers, 1988

2. 张连文, 郭海鹏. 贝叶斯网引论：北京：科学出版社, 2006

(刘大有　贾海洋)

bijiben jisuanji

笔记本计算机(notebook computer)　一种小型、便携式、适合于在移动环境中使用的**个人计算机**(PC)。笔记本计算机的平面尺寸与 A4 复印纸的尺寸相仿,其厚度在 3 cm 左右。打开笔记本计算机的顶盖,即露出操作键盘,在顶盖内部装有平板式液晶显示器,整台计算机的重量,连同机内电池在内,一般在 2 kg 左右。笔记本计算机的基本配置包括中央处理器、存储器、硬磁盘、光碟、显示器、键盘、定位设备和电源等。它和台式计算机在功能上没有什么差别,只是它的集成度更高,功耗更低,体积更小,重量更轻,功能日趋强大,相同配置时价格高于台式计算机。随着计算机制造业整体技术的不断提高和创新,笔记本计算机更加微型化和轻型化,售价不断降低,应用不断普及。

按用途笔记本计算机一般分为 4 类：商务型、时尚型、多媒体应用型、特殊用途型。商务型的特征为移动性强、电池续航时间长;时尚型外观特异,也有适合商务使用的时尚型笔记本计算机;多媒体应用型的笔记本计算机融合了强大的图形及多媒体处理能力,又兼顾移动性,配有高档独立显卡,较大的屏幕;特殊用途的笔记本计算机是服务于专业人士,可以在酷暑、严寒、低气压、工业及军用等恶劣环境下使用,大多较笨重。

笔记本计算机是便携式计算机的一种,便携式计算机分为膝上型、笔记本型和掌上型 3 类。1985 年膝上型计算机开始进入市场,它可以看成是台式计算机小型化以后的直接产物。它的外形像一个小型的公文包,功能齐全,接口丰富。用平板式显示器替代了台式机的阴极射线管显示器,用电池(一般为可充电电池)供电,电池的使用时间不到 1 h。连同电池在内,整台机器的重量为 5 ~ 10 kg。膝上型计算机经常在流动环境中使用,没有桌子时,可坐下来放在膝上使用,所以称为膝上型计算机。1988 年,日本 NEC 公司推出了一种设计独特的、称为 Ultraline 的膝上型计算机,其大小和一个 A4 纸张尺寸的笔记本相当,具备 IBM PC AT 机的功能,重量不到 2.3 kg,宣告了笔记本计算机的诞生。

笔记本计算机的技术特点

笔记本计算机的主要技术特点体现在以下几个方面：

(1) **中央处理器**(以下简称处理器)　笔记本计算机的核心部件之一,是选择笔记本计算机关注的焦点。处理器大都采用 Intel x86 系列的微处理器,早期也有少量采用 Motorola 公司生产的 68000 系列或 Power PC 系列的**微处理器**,或其他精简指令集计算机(RISC)处理器。处理器围绕主频、多核、二级和三级缓存容量、前端总线频率、功耗和价格等指标,形成了不同半导体工艺、不同微体系架构的低功耗处理器系列产品。目前,处理器都使用 Intel x86 体系架构,产品以 Intel 公司为主,其次是 AMD 公司,少量低端的用威盛公司的处理器。处理器都使用低功耗设计技术,以 Intel 系列的微处理器为

例，Intel 公司在便携式计算机用的处理器中，增加了能源管理技术。这种技术以 Intel 公司特有的系统管理模式（SMM）为基础，在中央处理器中增设了不可屏蔽的系统管理中断（SMI）以及与之相对应的复苏（resume）指令和相应的系统管理程序，可以提供透明于操作系统和应用程序的电源管理功能。可实现对中央处理器、主存、外围设备工作状态的控制，使之处于全工作状态、半工作状态或休眠状态（即闲置状态）。这些状态的控制均独立于操作系统和应用程序，状态间的切换可瞬间完成，既达到预期的节电效果，又不给系统程序员和应用程序员带来任何麻烦。

（2）显示屏　笔记本计算机一般采用平板式液晶显示器，它是笔记本计算机中成本最高的部件，占成本的 1/4 左右。1993 年以前，一般采用单色显示器，少数采用无源单扫描矩阵彩色显示器，色彩效果不理想，对比度和亮度较小，刷新时间长，画面移动时会留下踪迹。1993 年以后，开始采用双扫描技术，把屏幕分成两个可以同时刷新的部分，彩色较鲜艳，但是，视角较小。后来，又出现了有源矩阵彩显（用薄膜晶体管 TFT 显示），TFT 有出色的色彩饱和度、还原能力和更高的对比度，太阳下依然看得非常清楚，缺点是比较耗电，而且成本也较高。最近几年又出现了 LED 显示器，其显示屏是由发光二极管排列组成的显示器件。它采用低电压扫描驱动，具有耗电少、使用寿命长、成本低、亮度高、故障少、视角大、可视距离远等特点。与 LCD 显示器相比，LED 在亮度、功耗、可视角度和刷新速率等方面，都更具优势。目前，笔记本计算机的屏幕分辨率已经可与台式计算机相比。此外，笔记本计算机一般都备有连接台式计算机显示屏和投影仪的接口。

（3）存储器　分主存和外存。主存芯片从早期的**动态随机存取存储器**（DRAM）或**静态随机存取存储器**（SRAM），升级到了目前的 1.333 GHz 的 DDR3，主存芯片在集成度，存取速度等方面都有数量级的提高。主板中一般采用二个紧凑外形双列直插内存模块（SODIMM）来安装主存，最大容量可达到 8 GB。在容量达几个 MB 的处理器片内三级缓存支持下，存储系统的性能有了很大提高。外存目前用通用串行总线（USB）接口的 U 盘替代了早期的 3.5 in 的**软磁盘驱动器**。硬磁盘采用 2.5 in 或 1.8 in 的小型磁盘，厚度分 9.5 mm 和 12.5 mm 两种，具有体积小，功耗低的特点。笔记本磁盘普遍采用磁阻磁头（MR）技术或扩展磁阻磁头（MRX）技术，以极高的密度记录数据，增加了磁盘容量、提高了数据吞吐率，同时还能减少磁头数目和磁盘尺寸，提高磁盘的可靠性、抗干扰、抗振动性能。近年来半导体固态盘（参见**固态盘**）以其在读写速度、存取时间、功耗、体积、重量、耐冲击力等方面的优势，作为硬磁盘的替代物进入便携机市场。但是，固态盘在存储容量和价格方面仍不如硬磁盘。近年来，CD 或 DVD 光碟，不再作为外接扩充件，而是集成到笔记本计算机内。

（4）显卡　显卡是笔记本计算机选型的主要考虑因素之一。显卡分为集成显卡和独立显卡两大类，独立显卡性能优于集成显卡。集成显卡是将显示芯片、显存及其相关电路与主板融为一体，一些主板集成的显卡在主板上单独安装了容量较小的显存。独立显卡是指将显示芯片、显存及其相关电路做在一块电路板上，插入主板的扩充槽，一般不占用系统内存，比集成显卡能够得到更好的显示效果和性能，容易进行显卡的硬件升级。高端独立显卡的图形处理器的图形图像处理能力和显示存储器的速度分别优于笔记本计算机的中央处理器（CPU）和主存，对高端多媒体应用可获得很好的显示效果。

（5）电源　一般笔记本计算机既可用电池供电，也可用交流市电供电。电池一般用可充电电池，早期采用镍镉电池，每次充电前必须先放电，使用不方便。后来使用镍氢电池，它无记忆效应，使用较方便，而且单位重量的电量较大。其后出现锂离子电池，它的续航时间长，但价格比镍氢电池高。在同样重量情况下，上述 3 种电池的续航时间比为1∶1.2∶1.9。锂离子电池 1 次续航时间为 4～6 h。为了延长电池的续航时间，笔记本计算机普遍采用智能电源管理技术，如在不需要的情况下降低中央处理机的运行速度，降低总线工作频率，减低屏幕亮度，暂停闲置的硬磁盘驱动器，关闭未使用的通信端口的电源等。

（6）外壳　笔记本计算机最直接的保护体，也是影响其散热效果、重量、美观度的重要因素。笔记本计算机常见的外壳用料有：合金外壳，又分铝镁合金与钛合金；塑料外壳，又分碳纤维、聚碳酸酯 PC 和 ABS 工程塑料合金。一般而言，合金外壳在导热、坚固度、电磁屏蔽性、重量等方面比塑料外壳有优势，尤以钛合金外壳性能最佳。但是，钛合金外壳成本十分昂贵，仅用于高端产品。ABS 工程塑料合金由于成本低，尽管其重量和导热性能欠佳，仍被大部分笔记本计算机厂商所采用。

（7）输入设备　笔记本计算机底座上集成有嵌入式键盘和一套定位设备。定位设备用来替代鼠标

器(参见**鼠标**),早期一般使用轨迹球(trackball)作为定位设备,现在较为流行的是触控板(touchpad)与指点杆(pointing stick)。在高端笔记本计算机中,也开始使用采用多点触控技术(参见**多点触控**)的多点触控板。笔记本计算机也可通过USB端口或PS/2端口外接有线鼠标器,或者通过无线连接接入无线鼠标器。

(8)网络通信接口　一般笔记本计算机都配有10/100/1000 Mbps **局域网**接口。为了方便移动上网,在无线局域网的无线覆盖区,配有内置IEEE 802.11b/g/n **无线局域网**接口的笔记本计算机可直接接入无线局域网。无内置无线局域网接口的笔记本计算机可以通过USB接口的无线网卡接入无线局域网。此外,无线广域网覆盖的地区,可通过USB等接口的特定型号的无线上网卡,接入相应无线网络提供商的广域网实现上网。

笔记本计算机的发展趋势

笔记本计算机将继续沿着更轻、更薄、更廉价、性能和电池续航能力更强,使用更多样化的方向发展。笔记本计算机在配置和性能上已经逼近台式机,使用与台式机相同的操作系统和软件,而它的便携性和移动特点却是台式机无可比拟的。2005年以后,美国和日本等发达国家笔记本计算机的销量已超过了台式机。国内笔记本计算机市场增长与发达国家相比有些滞后,但是发展势头良好。国内笔记本计算机销量占整个个人计算机的份额已经从2000年的6%上升到了2009年的40%,呈现逐年增长的势头。近年来出现了以上网、电子邮件收发、音视频播放为主要用途的轻便型、小屏幕、便于无线上网的低配置笔记本计算机,称为上网本(netbook),以及**平板计算机**等产品,分流了部分笔记本计算机的市场,对笔记本计算机的市场份额产生了一定的负面影响。但是,从个人计算机发展总趋势来看,笔记本计算机成为市场主流并取代台式机的现有地位是一种必然趋势。

参考文献

1. Ralston A, Reilly E D, ed. Encyclopedia of computer science. 3rd ed. New York: IEEE Press, 1993

2. http://baike.baidu.com/view/7690.html? tp = 9_01

（韩承德）

bianji chengxu

编辑程序(editor)　对文件内容进行增加、删除及修改等操作的程序。它也可以用于创建、清除和复制文件本身。被编辑文件的基本元素可以是字符、文字、表格、图形及图像等。编辑对象是字符、文字和表格的编辑程序称为**正文编辑程序**,它常用于源程序的编辑。除字符、文字外,还可编辑图形的编辑程序称为**图形编辑程序**。以图像为主要编辑对象的编辑程序称为**图像编辑程序**。

编辑程序的用户界面包括输入、输出设备和交互语言。用户通过输入设备选择要编辑的文件、输入编辑命令和编辑元素;通过输出设备获取被编辑的元素和编辑的结果;交互语言定义规则和设施以使用户与编辑程序进行交互。编辑过程即是在用户编辑命令控制下对文件进行创建、删除、修改、移动、复制及显示打印的过程。

编辑程序的逻辑结构可分为输入、命令解释和输出三个部件。输入部件接受输入设备送入的字符、保存命令行及对功能信号进行译码;命令解释部件分析输入部件送来的信息,识别、解释相应的编辑命令、执行编辑动作;输出部件将编辑的结果写到输出设备上。

编辑程序在20世纪60年代即已出现,几十年来,随着计算机应用领域的拓广,编辑程序的功能不断增强。以正文编辑程序为例,就视域而言,从行编辑扩展到全屏幕编辑,又扩展到多窗口编辑;就编辑的正文结构而言,已从“无结构”扩展到“有结构”,如语法制导编辑;就处理能力而言,已不仅限于对文件内容的增、删和改,还提供诸如拼写检查、预浏览等创建和打印高质量文档的能力,如字处理程序。随着编辑程序功能的不断扩大,可望应用手写体识别、语音识别和图像识别等多媒体技术以使得人机界面变得尽可能丰富、自然和直觉。

参考文献

Beck L L. System software: an introduction to systems programming. 2nd ed. Addison-Wesley, 1990

（张素琴）

bianjiema jishu

编解码技术(coding technology)　针对特定的应用和目的,采用相应的规则和码型(code)对数据进行变换或处理的技术。将源数据转换为目标数据的过程称为编码(coding),其逆过程称为解码(decoding)。

编码通常可以用于模拟信息的数字表示(如麦克风采集的模拟语音信号等)、数据压缩、数据加

密、数据传输的检错和纠错，甚至有网络编码等。根据使用的范围，可以将编码技术应用于两个领域：

（1）信源编码（source coding） 针对各种信息源及不同目的所进行的编码。传统的信源编码主要考虑信息的表示，例如早期的莫尔斯电码、英文字符的 ASCII 编码等，主要完成英文信息的二进制表示。现代数据通信系统中常见的信源编码大部分与数据压缩有关，通过对原始数据的有损或无损压缩编码，可以极大地降低数据量，节省数据存储和传输的开销；例如：常用的无损压缩编码方式有 Huffman 编码、算术编码、L-Z 编码等；有损压缩编码方式有音乐的 MP3 编码、视频流的 H. 264 编码等。此外，数据加密也可以是一类特殊的信源编码技术。

（2）信道编码（channel coding） 其主要是选择特定的码型和编码规则（code）对需要传输的数据进行编码，使其可以对传输信道带来的错误进行检测（error detection），甚至纠正传输错误（error correction）。通常检错或纠错的效果越好，其编码算法的成本就越高，因此，信道编码经常是编码成本与性能的平衡折中。也就是说，编码算法的选择与数据信道的传输误码率（error rate of transmission）有关，与特定的应用要求有关。

信源编码和信道编码有时可以联合设计并优化。例如，互联网上常用的 ZIP 数据压缩系统，它不但具有强大的文件压缩功能，也同时具有文件传输检错功能，可以发现 ZIP 文件传输过程中的错误。

参考文献

1. 姚万生. 计算机网络原理与技术. 哈尔滨：哈尔滨工业大学出版社，2007

2. 胡道元. 计算机网络（高级）. 北京：清华大学出版社，1999

（周杰 张凌）

bianyi chengxu

编译程序（compiler） 将**高级语言**书写的程序翻译成等价的**机器语言**程序或汇编语言程序的处理系统。

编译程序以高级语言书写的程序作为输入，称之为***源程序***；而以机器语言或汇编语言表示的程序作为输出，称之为***目标程序***；其最终任务是产生一个可在具体计算机上执行的目标程序。执行目标程序将会按照用户在源程序中所规定的意图，加工初始数据，算出所需的结果，完成所希望的加工任务。源程序中的每个语句与目标程序中的指令通常是一多对应关系，所以编译程序的实现算法较为复杂，但它可以产生高效运行的目标程序，因此编译程序更适合于翻译那些规模较大、结构较复杂、运行时间较长的大型应用程序。

编译程序必须分析源程序，然后综合成目标程序。为此，编译程序要在分析阶段建立符号表、常数表和中间语言程序等数据结构，以便在分析和综合时引用和加工（图 1）。源程序的分析是经过词法分析、语法分析和语义分析三个步骤完成的，目的是检查源程序的语法和语义的正确性，并把源程序分

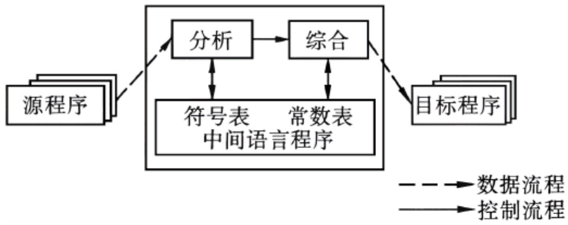


图 1 编译程序工作过程示意图

解成一系列的基本组成成分。目标程序的综合通常包括存储分配、代码优化、代码生成等几个步骤，目的是为源程序的常数、变量、数组等数据结构分配存储空间，重新组织分析阶段产生的基本组成成分，将其综合成高效运行的可执行目标程序。

编译程序在逻辑上由分析和综合两大部分组成，并可进一步细分为**词法分析**、**语法分析**、语义分析、存储分配、代码优化和代码生成 6 个相继的逻辑步骤。具体设计和实现编译程序时，通常是按照从头到尾扫视源程序（或其等价的中间语言程序）的遍数来规划编译程序的结构，安排相关逻辑步骤的工作。每一遍可以按顺序执行方式或并行调用方式，完成一个或相连几个逻辑步骤的工作。例如，可以把词法分析作为第一遍；语法分析和语义分析作为第二遍；存储分配和代码优化作为第三遍；代码生成作为第四遍。反之，为了适应较小的内存空间或提高目标程序质量，也可以把一个逻辑步骤的工作分为几遍去完成。例如，代码优化可划分为代码优化准备和实际代码优化两遍来完成。编译程序采用多少遍的编译结构，应根据机器的规模、程序语言的繁简、编译程序的功能、目标程序的质量、设计人员的多少等具体情况而定。一遍编译程序是一种极端的情况，整个编译程序同时驻留在内存，彼此之间采用调用转接方式连接在一起工作（图 2）。当语法分析程序需要新符号时，它就调用词法分析程序；当它识别出某一语法结构时，它就调用语义分析程序。语义分析程序对识别出的结构进行语义检查，并调

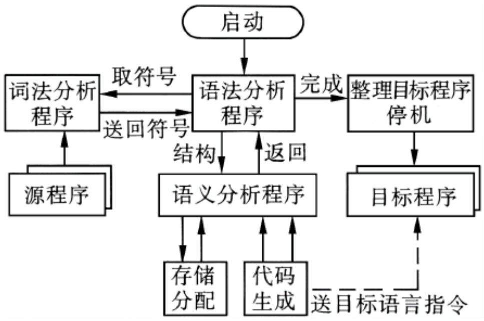


图 2 一遍编译程序

用“存储分配”和“**代码生成**”程序生成相应的目标语言的指令序列。

随着高级语言在形式化、结构化、智能化和可视化等方面的发展，作为实现相应语言功能的编译程序，也随之向自动程序设计和可视化程序设计的方向发展，为用户提供更加理想的程序设计工具。

参考文献

1. Gries D. Compiler construction for digital computer. New York: Wiley, 1971

2. Aho A V. Compilers principles, techniques and tools. New York: Wesley, 1986 （曹东启）

bianyi chengxu de bianyi chengxu

编译程序的编译程序（compiler-compiler） 产生编译程序的编译程序。它接受用某一适当的表示体系描述的某一语言类中任一语言 A 的词法规则、语法规则、语义规则和（或）代码生成规则，并从这些描述产生出用目标语言 B 写的关于语言 A 的编译程序的全部或部分。编译程序的编译程序又称为*编译程序的生成程序*。

通常，编译程序分成词法分析程序、语法分析程序、语义分析程序、代码生成程序等若干部件。这些部件可以用一个总的编译程序的编译程序的不同部分来生成，也可以分别用不同的专用生成程序来生成。这些专用生成程序包括词法分析程序的生成程序，语法分析程序的生成程序和代码生成程序的生成程序等。

*词法分析程序的生成程序*接受以正则文法或其他类似文法描述的单词，构造一个有限状态自动机，由此生成一个词法分析程序。

*语法分析程序的生成程序*接受以上下文无关文法的形式描述的源语言的语法，生成一个语法分析程序。各种语法分析程序的生成程序随实现语言、语法分析算法的原理以及源程序中语法错误处理能力的不同而异。如采用 LR 的语法分析程序的生成程序 YACC 和采用递归下降法的语法分析程序的生成程序 LLgen。

语义分析程序、中间代码生成程序和目标代码生成程序的生成程序的设计与实现与形式化的语义描述紧密相关。语义描述形式化技术相当困难，目前大多数生成程序中语义描述还是采用非形式化，它们的基本思想是为源语言的上下文无关文法的语法符号或产生式配以翻译子程序（语义动作或语义子程序）。

现有不少性能很好的编译程序的编译程序，如词法分析程序的生成程序 LEX，语法分析程序的生成程序 YACC 和 LLgen，它们都显著提高了编译程序的开发效率。

参考文献

1. 陈火旺，钱家骅，孙永强. 程序设计语言编译原理. 2 版. 北京：国防工业出版社，1984

2. Aho A V, Sethi R, Ullman J D. Compilers principles, techniques and tools. Addison-Wesley, 1986 （徐永森）

bianliang

变量（variable） 具有“**值**”这一属性、且可视需要对其值进行更新的计算对象。变量可用作诸如世界人口、某人年龄、当前天气等具有状态的现实对象的模型。

有两种变量，一种可在程序内部创建并使用，且可通过赋值更新其值，生存期较短；另一种如文件、**数据库**等，生存期较长，其存在和特定程序无关，其更新是有选择的。

程序设计语言中的变量和数学变量不同，前者可以更新其值，后者代表一个固定但未知的值，该值并不能改变。

随着语言不同，有的变量无类型，有的变量有类型。变量的具体类型规定了它的取值范围，如果由于某种原因被赋予的值超出了这一范围，就能被发现，所以有类型的变量较为安全。（参见**数据类型**）。

参考文献

Watt D A. Programming language concepts and paradigms. Prentice Hall, 1990 （徐家福）

bianxing donghua

变形动画（deformation animation） 采用变

形技术作为运动控制的动画技术。变形可以是二维的或三维的，可以是几何的或图像的。传统动画的一个显著特点是赋予每个角色以个性，并以形状变形来渲染某些夸张的效果。虽然传统动画的有些夸张效果用三维动画还很难实现，但**计算机动画**的研究者们已在形状变化方面做了不少出色的工作，并在电视、电影、广告和音乐电视（MTV）中得到了广泛的应用。大部分变形方法与物体的表示有密切关系，如通过移动物体的顶点或控制顶点来对物体进行变形。为了使变形方法能很好地结合到造型和动画系统中，人们提出了许多与物体表示无关的变形方法。

变形动画可分为形态变换和空间变形两大类。形态变换是指将一给定的源数字图像或几何对象 S 光滑连续地变换到目标数字图像或几何对象 T。在这种光滑过渡中，中间帧应既具有 S 的特征，也具有 T 的特征。S 和 T 的拓扑可以相同也可以不同。形态变换通常需要动画师指定源处理对象和目标处理对象之间特征的对应关系，当然，这种对应关系也可以由系统自动求得。空间变形指将单个几何对象的形状做某种扭曲、变形，使它变换到动画师所需的形状。在这种变化中，几何对象的拓扑关系保持不变。空间变形更具有某种随意性，所以也常称为自由变形。空间变形包括与物体有关的变形和与物体无关的变形。

参考文献

1. Parent R. Computer animation: algorithms and techniques. 2nd ed. San Francisco, CA: Morgan Kaufmann, 2007

2. Kerlow IV. The art of 3D computer animation and effects. 4th ed. Wiley Publishing, 2009

（金小刚）

bianxieshi meiti bofangqi

便携式媒体播放器（portable media player, PMP） 一类可以存储和播放音频、图像、视频和文档等数字媒体的便携式消费类电子设备。该设备通常用内置或外接的**快闪存储器**或微硬盘作存储器，并配有单色或彩色液晶显示屏（参见**显示器**），便于随身携带。用盒带来录制和播放音、视频的模拟式媒体播放器，已逐渐被淘汰，不在讨论范围内。

最早的数字式媒体播放器（PMP）可追溯到 1979 年英国科学家研制成功的数字式音乐播放器的实验室原型 IXI，直到 1996 年便携式媒体播放器的批量产品才上市，最早流行的是 MP3 随身听。目前，市场上常见的媒体播放器有 MP3 随身听、MP4 随身看、iPod 等，**数字相机**、视频录相机、电子书等由于其具有回放功能，也有人将之归入 PMP。

便携式媒体播放器最基本的功能是为用户提供数字视频和数字音频的体验。除此之外，目前很多 PMP 还支持视频录像、摄像、数字相机伴侣、数字收录音、移动存储等多项功能。

处理能力、存储容量、能耗、设计和用户界面、支持的文件格式、屏幕大小和分辨率、特色功能、价格等是衡量 PMP 功能的主要考虑因素。

处理器是便携式媒体播放器的核心部件之一。便携式媒体播放器的处理器种类较多，源自不同的架构，在性能、接口、功耗等方面差异很大。例如，Intel 和 AMD 提供的是通用处理器，能支持丰富的音视频格式，依赖软件解码，其扩展性强，可以按需增加软件编解码器，但是功耗较大。另一类采用**中央处理器**（CPU）与**数字信号处理器**（DSP）的协同处理方式，例如，TI、PHLIP 的一些产品。CPU 用来处理系统的通用功能，如**操作系统**和用户接口等；DSP 完成音视频解码，通过软件升级也可支持新的编解码器，可实现高性能、高复杂度的视频编解码器。再一类是采用嵌入式处理器和专用的硬件编解码器，其功耗低、可扩展性差。

由于视频格式种类繁多，任何一台 PMP 都不可能支持所有格式，例如大部分 PMP 都不支持互联网上流行的 RM、RMVB 格式。对于不支持的格式，可以先在 PC 上进行转码，然后再在 PMP 上播放。某些 PMP 自带格式转换的软件。

功耗是便携式设备的关键指标。与手机不同，PMP 是一种个人娱乐产品，不存在待机的问题，所以，电源管理的核心是如何降低产品在运行时的功耗。目前，有些产品的电池续航时间音频达 24 小时，视频达 4 小时。

特色功能诸如对无线网络、即时通信和全球定位系统（GPS）的支持，使 PMP 提供随时随地上网、聊天，欣赏音乐或视频，进行出行导航等功能，吸引了更多消费群体。

内容是 PMP 扩大用户群体的源泉，多媒体音视频的开发和版权是急待解决的问题。

PMP 产品中，MP4 播放器是中国创造的新名词，指一个能够播放 MPEG-4 文件（参见**运动图像的压缩编码标准**）的设备，也可以叫做个人视频播放器（PVP）。

Apple 公司的 iPod 产品系列的屏幕、创新的界面设计和特色功能、产品更新换代速度等引人注目，在同类产品市场上始终占据首位。PMP 的产品正处于迅速发展期，随着移动多媒体标准的颁布、移动数字电视网络的普及，移动多媒体处于蓬勃发展时期，正在改变着人们的生活方式，为便携式媒体播放器的发展展示了良好前景。　　（韩承德）

biaoji yuyan

标记语言（markup language）　一种用来对文本进行注释的语言，其提供的注释与原始文本在语法上是可区别的，也称**置标语言**。标记的思想起源于编者或印刷工人关于手稿的修订说明或排版指令。到 20 世纪 60 年代，随着计算机文字处理系统的发展，诞生了多种标记语言，包括 IBM 的通用标记语言（generalized markup language，GML）。标准通用标记语言（standard generalized markup language，SGML）就是以 GML 为蓝本制定的，于 1986 年成为国际标准化组织的一个标准（ISO 8879：1986）。从 1990 年起，随着万维网的发展，超文本标记语言（hypertext markup language，HTML）和可扩展标记语言（extensible markup language，XML）逐渐成为使用最为广泛的标记语言，对万维网上信息传播和数据交换发挥了重要作用。

标记通常可以分为三种：呈现型、过程型和描述型。呈现型标记用来指示或说明相关文本的呈现效果，比如字体和颜色等外观。过程型标记为处理文本的程序提供相关指令，旨在实现某项功能。描述型标记旨在描述文本的内容或结构，而不是文本的呈现外观或样式。随着技术的发展，这三种类型之间的界线逐渐模糊，比如，表达呈现效果或者实现某项功能的标记也会采用 XML 来描述。

标准通用标记语言（SGML）的基本思想是将文档的内容结构与样式分开，推崇描述型标记，提倡标记的严格性和使用的灵活性。需要指出的是，SGML 是一种**元语言**，可以用来定义特定的标记语言。事实上，**超文本标记语言**（HTML）就是一种基于 SGML 的标记语言。作为一种特定的标记语言，HTML 只提供有限种标记，且注重于文本的呈现效果，难以满足万维网上交换数据的发展要求。**可扩展标记语言**（XML）比 SGML 简洁很多，同时继承了 SGML 的大部分优点，也是一种元语言，可以用来定义应用领域的标记。目前，XML 已成为万维网上数据表示的一种重要语言。

参考文献

Goldfarb C F, Prescod P. The XML handbook. 3rd ed. Prentice Hall, 2000　　（瞿裕忠）

biaoliangchang keshihua

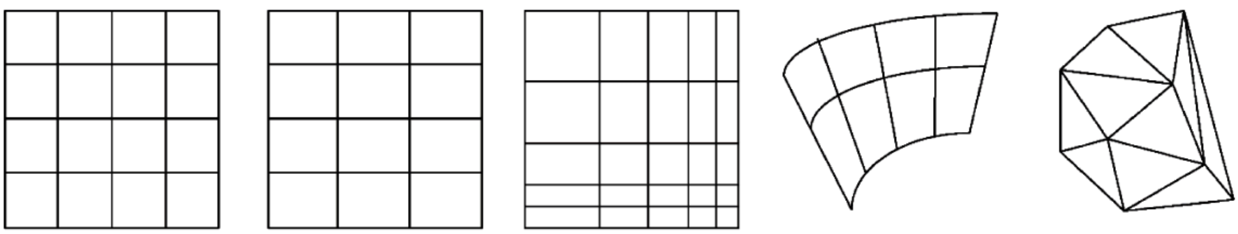
标量场可视化（scalar field visualization）

指通过计算、显示、跟踪和分析标量场（scalar field）的几何结构、空间分布、拓扑关系等内在特征的**可视化**方法。由于很多科学测量或者模拟数据都是以标量场的形式出现，对标量场的可视化是科学可视化研究的核心课题之一。

标量场的空间中每一点的属性都可以由一个单一数值（标量）来表示，表达为 f：D11，D11^{2}11^3。标量场也可以随时间变化，即时变标量场。常见的标量场包括温度场、压力场、势场等。

标量场既可以是二维表面，也可以是三维数据场。根据空间采样点的排布以及相互间的连接关系不同，标量场可以是高度结构化的线性网格或者非结构网格。在科学计算或者工程实践中，还有可能是由多个不同网格合并而成的一个三维标量场。图 1 所示是几类典型的二维网格。三维标量场也常被称为体数据。体数据中的单元称为体素（voxel），对应于二维图像的像素。每个体素对应于在三维空间中的网格格点上采样的数值。

最常见的标量场可视化方法包括颜色映射（color mapping）、轮廓法（contouring）以及高度图（height plot）。颜色映射的方法将每一标量数值与一种颜色相对应，可以通过建立一张以标量数值作



立方网格　各向异性直角线性网格　直角线性网格　曲线网格　不规则网格

图 1　典型的二维网格种类

为索引的颜色对照表的方式实现。更普遍的建立颜色对应关系的方法称为传递函数(transfer function),它可以是任何将标量数值映射到特定颜色的表达方式。对于颜色映射的可视化,选择合适的对应颜色非常重要,不合理的颜色方案将无法帮助解释标量场的特征,甚至产生错误的信息。轮廓法(contouring)是将标量场中数值等于某一指定阈值的点连接起来的可视化方法。地图上的等高线、天气预报图中的等温线都是典型的二维标量场的轮廓可视化的例子。多条等值轮廓线(或等值轮廓面)在标量场上分布的稀疏程度表示了相应标量场变化的快慢。二维标量场的轮廓线可以通过移动正方形(marching square)的方法获得。三维标量场的轮廓可视化即为等值面的提取和绘制。高度图(height plot)则是根据二维标量场数值的大小,将表面的高度在原几何面的法线方向做相应的提升。这样表面的高低起伏对应于二维标量场数值的大小和变化。

三维标量场也被称为三维体数据场(volumetric field),其主要可视化方法包括直接体绘制和等值面的提取与绘制。直接体绘制通过颜色映射,可以直接将三维标量场投影为二维图像。这种算法并不构作中间几何图元,而是由离散的三维数据场直接产生屏幕上的二维图像。选择三维标量场的颜色映射方案就是对体数据的直接体绘制设计传递函数的问题。如何设计合理的传递函数一直是可视化研究中的重要课题。等值面方法可以更好地表示特定曲面的特征和信息,但是与直接体绘制方法相比,丢失了指定等值面以外的数据场信息。另一方面,直接体绘制虽然显示了包括全部三维数据场的信息,但是由于数据之间的遮挡以及体绘制中的合成计算,特征之间可能发生干扰。如何通过选择合理的传递函数,使得体数据可视化最佳地揭示内在特征是一个很大的挑战。此外,三维标量场还可以通过设立切面(slicing)的方式对特定平面的信息可视化,这种方法在医学成像数据方面使用较多。

对标量场可视化的研究已经从最初的关注于效率问题,到更加注重对其内容的分析和交互处理上。如何可视化三维标量场中的不确定信息、选择高效合理的传递函数、比较多个标量场、处理 TB 乃至 PB 量级的标量场数据等都是具有高度挑战性的课题。此外多个空间上重合的标量场组成一个多变量场也是常见的情况,例如医学中 CT、MRI、PET 等多模式成像,科学模拟计算每个网格点上可能有多个不同的标量变量。对多变量标量场的可视化非常值得探索和研究。最新的工作还包括引入信息可视化的方法,分析处理标量场数据的可视化。

参考文献

1. Hansen C D, Johnson C R (eds.) The visualization handbook. Elsevier, 2005

2. Telea A C. Data Visualization: Principles and Practice. Wellesley MA: A K Peters, 2008

3. 唐泽圣,等. 三维数据场可视化. 北京:清华大学出版社,1999

(唐泽圣 袁晓如)

biao chuli yuyan

表处理语言(list processing language) 一种用于描述表及其处理的**程序设计语言**。表是一种特殊数据集,它是由若干个(包括零个)元素组成的序列。此种数据结构灵活、方便、通用。

表处理的基本操作有:将新元素插入表的指定位置中;访问表中任一元素;删除表中指定位置的元素;将两个表链接成一个新表;将一个表拆成两个表;判断指定表是否空。

第一个表处理语言是 1958 年由 A. Newell 等研制的 IPL-V。最典型的表处理语言是 1960 年由 J. McCarthy及其研究小组设计实现的 LISP。其他表处理语言还有 Slip, Pop-2 等。20 世纪 70 年代末以来出现了几十种 LISP 版本,并建立了国际标准 Comman LISP。在面向对象技术热潮推动下,1988 年美国学者将当时若干有代表性的 LISP 合并为 CLOS,使之成为 AI 领域重要的语言。

参考文献

1. Newell A. Information processing language-V Manual. Englewood Cliffs, NJ: Prentice-Hall,1961

2. Winston P H, Horn B. LISP. 3rd ed. Reading, MA: Addison-Wesley, 1989

3. Steele G L. Common LISP. 2nd ed. Bedford: Digital Equipment Coration, 1990

(郭浩志)

biaodashi

表达式(expression) 高级语言中用来指明求值对象与规则的基本语法成分。

表达式可以是简单的,也可以是复杂的,但一般都涉及参与计算的运算对象(或称为操作数),进行计算的运算符,也可以有指明求值次序的圆括号。

表达式的运算对象可以是无正负号常量,变量,函数命名符,或由圆括号括起来的另一层表达式。

变量可以是由单个标识符表记的整体变量,也可以是构造类型的成分变量(如数组的下标变量,记录的域变量等)。

运算符用于对运算对象的求值。若按参与运算的对象类型来分类,则可分为算术运算符、关系运算符、逻辑运算符及集合运算符,而从运算涉及的运算对象个数来看,也可分为单目运算符和双目运算符,双目运算符又可分为乘除运算符、加减运算符、关系运算符及逻辑运算符等。同一层表达式的各种运算符,一般按数学上的先乘除后加减的原则来定义它们的优先级。例如,**Ada 语言**中对运算符定义了 6 个优先级:乘幂(* *),绝对值(ABS),非(NOT)等单目运算符优先级最高,AND,OR,XOR 等逻辑运算符优先级最低。同一层表达式的计算一般从左向右进行,但优先级高的运算符先做。

表达式可分为**同构型表达式**和**混合型表达式**。同构型表达式要求其所有成分都属于同一类型,例如,ALGOL 60 中定义的三类表达式,即算术表达式、布尔表达式及命名表达式,均属于同构型表达式,但目前多数语言的表达式则为混合型表达式,例如,Modula-2,Ada 等语言,一个表达式中也许有不同类型的运算对象,它们通过类型转换来解决运算对象类型不一致问题。

有类型语言要求参与计算的运算对象、运算符是有类型的,因而其表达式也是有类型的。运算符的类型可以从运算符的符号不同来区分,例如"÷"为整数类型运算符,"/"为实数类型运算符,但当前不少程序语言(如 Ada,C++等)中,同一个运算符,其类型根据其运算对象类型的不同可以作不同的解释。例如,Ada 语言中的"="运算符,其类型根据其运算对象类型,可解释为整数相等比较,也可解释为实数相等比较,甚至可以是用户定义的某一类型的相等比较,这种现象称为运算符的一名多用。

参考文献

Ralston A,Reilly E D. Encyclopedia of computer science. 3rd ed. New York:Van Nostrand Reinhold,1992

(陈涵生)

biaomian anzhuang jishu

表面安装技术(surface mounting technology,SMT) 用**波峰焊**或再流焊技术,将无引线或短引线的片状元器件(亦称表面安装元器件)焊到印制电路板或其他基板上的一种工艺技术。表面安装是在高速自动贴装机上进行的,其过程包括贴装和焊接两步。表面安装与通孔插装不同,表面安装时的元器件面与焊接面均在印制电路板或基板的同一面上。表面安装具有组装密度高、印制电路板或基板占用面积小、电气性能好、抗干扰能力强、可靠性高、成本低及易实现自动化生产等优点。它涉及材料(基板材料与工艺材料)技术、组装技术(贴装、焊接及清洗等)、设计技术、测试技术、标准化、可靠性等多种交叉学科技术。

1966 年,美国 RCA 公司研制成功片式薄膜电阻,并用于微膜组件上。20 世纪 70 年代日本将其应用于民用电子产品。80 年代中期开始,我国电子行业开始引进、消化、吸收表面安装技术。进入 90 年代以后,表面安装技术作为**微组装技术**的一项关键技术,正处于蓬勃发展的新阶段。

表面安装技术包括表面安装元器件、表面安装工艺和表面安装设备三方面。表面安装技术的发展主要取决于表面安装元器件。

表面安装元器件可分为无源元器件、有源元器件、机电元器件和集成电路等几类。从外形上可分为矩形片式(扁平型)元器件和电极面焊型(圆柱型)元器件两大类。不论哪一类表面安装元器件,它们都应符合下列基本要求:元器件的尺寸、形状应符合标准化和自动化安装的要求;有足够的精度和一定的机械强度;具有良好的电气性能并耐有机溶剂洗涤等。

表面安装工艺有 4 种:①在印制电路板的一面全部贴装表面安装元器件的单面安装;②在印制电路板的两面全部贴装表面安装元器件的双面安装;③在单面安装印制电路板上又插装有引线元器件的单面混合安装;④在印制电路板的一面贴装表面安装元器件,另一面贴装表面安装元器件和插装有引线元器件的双面混合安装。表面安装元器件的贴装方法有顺序式、同时式、流水线式和顺序同时式 4 种。顺序式是将印制电路板装在 x-y 工作台上,把表面安装元器件逐个顺次贴装。同时式是用盒式包装或料斗进料,通过模板成批同时贴装。流水线式是使印制电路板沿传送轨道,通过各个贴装头,将表面安装元器件依次贴装到相应的位置上。表面安装的焊接技术有再流焊和波峰焊两大类。再流焊的优点是热应力小,特别适用于半导体片式器件的焊接,其加热方法有红外加热、光束加热、汽相加热和激光加热等。波峰焊一般用于混合安装的有引线元器件的焊接,其缺点是对片式元器件热应力大,对有的片式元器件不适用。

表面安装的设备包括用来印刷焊膏或贴片胶的丝网印刷机、自动贴装机、再流焊机(或波峰焊机)、清洗机及检测设备等,其中自动贴装机是表面安装技术中的关键设备。

目前表面安装技术研究的重点在于高密度、高速度、细间距、高可靠组装设计、基板制造、焊膏及检测等工艺技术。 (赵惇殳)

biao tuiyan fangfa

表推演方法(tableau method) 通过分析公式结构寻找矛盾从而证明公式不可满足性的推理方法。表推演方法根据输入公式构造一棵节点标有公式的树,如果树的每个分支上都有互相矛盾的公式,则表明输入公式是不可满足的。

以**一阶逻辑**中的表推演方法为例,进行推理时有非子句表推演方法和子句表推演两种方法。非子句表推演方法的输入是一阶逻辑公式,将输入公式标在树的根结点上。根据其顶层逻辑联结符的语义把一阶逻辑公式分为 α、β、γ 和 δ 四种类型(见表1)。针对这四种类型有相应的规则来扩展树。当树的某分支上包含一个 α 类型公式时,则为该分支叶结点生成儿子及后代结点,形成长度为 n 的链,子公式 α_1 至 α_n 分别标在新生成的 n 个结点上。当某分支包含一个 β 类型公式时,则为该分支叶结点生成 n 个儿子结点,子公式 β_1 至 β_n 分别标在新生成的 n 个儿子结点上。当某分支包含一个 γ 类型公式时,则为该分支叶结点生成唯一儿子结点,标注的公式为$\gamma_1(y)$,其中 y 是树中从未出现的变量。当某分支包含一个 δ 类型公式时,则生成唯一儿子结点,标注的公式为 $\delta_1(sko(x_1,\cdots,x_k))$,其中 sko 是树中从未出现的 Skolem 函数,$x_1,\cdots,x_k$ 是树中出现的全部自由变量。树的分支上只有处理 γ 类型公式的规则可以对同一公式多次应用,其余规则对其适用的公式只能应用一次。这种表推演方法可得到易读性强的证明过程,并且适合于非经典逻辑推理。子句表推演方法的输入是一阶逻辑公式的 Skolem 范式,即仅受全称量词限制的子句的集合。此时,将 T 标在树的根结点上,输入的子句均可用来扩展树。当使用某子句扩展树时,相当于先后将 γ 规则和 β 规则作用在该子句上。因此,树的非根结点上标注的是文字。基于这一特点,该方法引入各种提高推理效率的策略。在对不含等词的一阶逻辑问题推理方面,基于子句表推演方法的机器证明程序可以与基于归结方法的程序相媲美。

表1 一阶逻辑公式类型的划分

α 型: α	$\alpha_1,\cdots,\alpha_n$	β 型: β	$\beta_1,\cdots,\beta_n$
$\varphi_1\wedge\cdots\wedge\varphi_n$	$\varphi_1,\cdots,\varphi_n$	$\varphi_1\vee\cdots\vee\varphi_n$	$\varphi_1,\cdots,\varphi_n$
$\neg(\varphi_1\vee\cdots\vee\varphi_n)$	$\neg\varphi_1,\cdots,\neg\varphi_n$	$\neg(\varphi_1\wedge\cdots\wedge\varphi_n)$	$\neg\varphi_1,\cdots,\neg\varphi_n$
$\neg\neg\varphi_1$	φ_1		
¬false	true		
¬true	false		
γ 型: γ	γ_1	δ 型: δ	δ_1
$(\forall x)(\varphi(x))$	$\varphi(x)$	$\neg(\forall x)(\varphi(x))$	$\neg\varphi(x)$
$\neg(\exists x)(\varphi(x))$	$\neg\varphi(x)$	$(\exists x)(\varphi(x))$	$\varphi(x)$

目前表推演方法已被自然而有效地推广到了各种非经典逻辑。它的主要优点是避免了过分的范式要求,为经典逻辑和各种非经典逻辑提供了一种解析证明方法的统一框架。对标准表推演系统做适当扩充可自然地用于**描述逻辑**、**模态逻辑**、**时态逻辑**、**直觉主义逻辑**、**多值逻辑**等逻辑的推理之中,并为在证明搜索过程中使用启发式信息提供了可能。随着计算机科学对非经典逻辑兴趣的与日俱增,表推演方法也被越来越多的人所重视、熟知和使用。

参考文献

D'Agostino M, Gabbay D M, Hähnle R, et al. Handbook of tableau methods. Kluwer Academic Publishers, 1999
(欧阳丹彤 冯莎莎)

bingfa chengxu sheji

并发程序设计(concurrent programming) 一种**程序设计**方法,按这种方法设计时,一个程序由若干个可同时执行的程序模块组成,这些可同时执行的程序模块称**进程**。严格地说,这里指的是进程描述,非进程本身。进程由数据和有关的语句或命令序列组成。组成一个程序的多个进程可以多台处理机并行地执行,也可以在一台处理机上交叉地执行。采用并发程序设计可以提高计算机系统效率,缩短程序执行时间。对于多机处理系统,上述结论是明显的,对于单处理机系统也是正确的。

例如一个从磁盘读入数据、加工后打印输出的程序,采用并发程序设计后它由三个进程组成:读盘进程,加工进程和打印进程。三个进程可轮流地占用处理机执行程序。在打印进程启动打印机后,在打印机工作期间,打印进程就无须占用处理机而处理机可被加工进程占用来加工数据或被读盘进程占

用来控制读入数据。这样就增加了外部设备和主机操作的并行度，从而提高了系统效率，缩短了程序的执行时间。

并发程序设计是在**多道程序设计**的基础上发展起来的。1968年，E. W. Dijkstra首先引入了并发程序设计的概念并研究了有关的同步和死锁问题。70年代人们开始将有关并发程序设计的概念引入程序设计语言中，出现了并发PASCAL、Modula和Ada等并发程序设计语言。

采用并发程序设计时必须处理好进程同步和**死锁**。所谓进程同步是指有共享数据的若干个进程对它们的访问共享数据的一种约制。例如P1和P2分别是将数据送入缓冲B和从缓冲B读出数据的两个进程。如果P1和P2不按先送后读的次序运行，那么可能一个数据被P2两次读出或者一个数据未读出时就被新读入的数据冲掉了。为了防止产生这样的错误，操作系统为并发程序设计提供了同步机制。**PV操作**就是具有代表性的同步机制。P操作和V操作是操作系统提供的两条原语。所谓**原语**是指它的执行是不会被打断的。执行P操作P(S)时信号量S之值减1，若结果不为负则P(S)执行完毕，否则执行P操作的进程暂时停止执行等待释放。执行V操作V(S)时，信号量S之值加1，若结果不大于0则释放一个因执行P(S)而等待的进程。有了PV操作后上面的例子所产生的问题就不难解决了。在上例中我们定义两个信号量S1和S2，它们的初始值分别为1和0。进程P1在送入数据前执行P操作P(S1)，在送入数据后执行V操作V(S2)。进程P2在读取数据前先执行P操作P(S2)，在读出数据后执行V操作V(S1)。当P1送入一数据后信号量S1之值变为0，在该数据读出后S1之值才变为1，因此在前面一数未读出前后面一数不会送入。这就保证了进程P1和P2的同步。

死锁给并发程序设计带来了另一个问题。死锁是指进程间因为竞争资源而产生的无休止的等待。例如有两个进程P1和P2，P1占有资源A而申请资源B，P2占有资源B而申请资源A。显然它们两个都不能得到所需的资源而无休止地等待下去。解决死锁问题有两种途径，一是设计防止死锁的资源调度算法，另一途径是检测死锁使得当死锁发生时能及时发现并进行排除。

参考文献

Hansen P B. 并发程序的系统结构. 杨芙清，等译. 北京：国防工业出版社，1982

（孙钟秀）

bingfa chengxu sheji yuyan

并发程序设计语言（concurrent programming language） 用于进行**并发程序设计**的**程序设计语言**。它的主要特征是引入了进程描述，因此，用它编写的程序指明了若干个可以同时执行的进程。此外这类语言还提供了进程同步、进程通信和进程产生等功能。

进程同步功能主要靠系统提供的同步机制来实现。进程同步机制有PV操作、管程等。管程由一组可供所有进程访问的数据及有关的操作组成。只有引用管程操作是互斥地被引用，显然管程中提供为多个进程访问的数据是互斥地被各个进程访问的，这就是实现了同步控制。

进程首先的功能是指在程序执行的过程中一个进程可以产生出一个新的进程。与之相对应，系统也就提供撤销进程的功能。一般说，只有创建进程的进程才能撤销它所创建的进程。

迄今为止已有多种并发程序设计语言，如并发PASCAL、并发C、Modula-2、Ada和Occam等。

（孙钟秀）

bingfa kongzhi

并发控制（concurrency control） **数据库管理系统**（DBMS）中协调多个事务的并发执行的机制和过程。并发执行指多个事务在共同时间段内同时执行。

当多个事务并发执行时，即使各个事务分别能保证事务执行的正确性和数据库状态的一致性，但多个事务的交叉调度所形成的事务之间的相互影响可能导致不一致。常见的不一致性问题包括丢失更新、读脏数据、不一致的分析等。如果一个多事务并发调度与同一事务集合的某个串行调度执行过程等价，则称这个调度是可串行化的。可串行化的调度能保证事务执行的正确性和数据库状态的一致性。

DBMS采用的保证事务调度可串行化的主要方法包括封锁方法、时间戳方法、基于有效性确认的方法。其中最常用的是封锁方法，其基本思想是，事务在对任何数据库元素进行读写操作之前都必须申请对该数据库元素的适当封锁，系统通过不同类型锁之间的冲突来控制不同事务对相同数据库元素的读

写,若系统各事务对封锁的申请和释放遵守两阶段封锁协议,则能保证事务调度是可串行化的。

参考文献

1. Date C J. 数据库系统导论. 孟小峰,王珊,等译. 北京：机械工业出版社,2000

2. Garcia-MolinaH, Ullman D, Widom J. 数据库系统实现. 杨冬青,唐世渭,徐其钧,等译. 北京：机械工业出版社,2001

（杨冬青）

bingfa moxing

并发模型（models of concurrency） 描述并发系统行为的数学模型。

若一个系统内部发生的两个事件之间没有因果关系,则称此两个事件是并发的。因果关系不等于事件先后关系,有因果关系者必有先后关系,反之则不一定。存在并发事件的系统称为并发系统。例如操作系统是一个并发系统,人类社会也是一个并发系统。

并发概念由 C. A. Petri 于 1962 年首创。他的并发模型严格遵守并发即无因果联系的思想,用此模型描述的系统不含统一的时钟。此外还有另一种并发概念,例如 R. Milner 认为：若一个系统内部的两个事件可以按任意次序发生,则称此两个事件是并发的。习惯上称前一种并发为真并发,称后一种并发为交叠式并发。前一种并发概念的描述能力强于后一种并发概念的描述能力,但是在程序设计的大多数场合,后一种并发概念也够用了。

并发模型可分为两个层次：描述性的并发模型和语义性的并发模型。

描述性的并发模型 通常是一个形式系统,可以描述并发系统的行为。这种模型大体上有 4 类。

归约模型：**λ 演算**、项重写系统、图重写系统等都是。这类系统中的事件就是归约。它们缺少进程和通信的概念,不能用于分布式系统。

逻辑模型：时序逻辑、动态逻辑、**线性逻辑**等都是。这类系统中的事件就是逻辑推导。它们能够在进程上作推理,但是缺少有效的通信手段,因此也不适用于分布式系统。

进程代数模型：**通信顺序进程**（CSP）、**通信系统演算**（CCS）等都是。这类系统以进程及进程间的通信为主要描述对象。系统中的事件就是进程通信,它们特别适合于描述分布式系统。

佩特里网模型：EN 系统、条件或事件系统、位置或变迁系统、有色佩特里网系统等都是。这类系统不直接描述进程和通信,但是可通过并发网络流表示进程和通信。系统中的事件就是事件节点的点火。这类模型不仅用于描述程序系统,也用于描述社会系统。

语义性的并发模型 第一类并发模型通常以指称语义中的幂域理论为基础。第二类并发模型常采用 Kripke 的可能世界理论。第三类并发模型也采用幂域理论,但采用不同的对象作为域的元素。例如针对同一个 CSP,M. Broy 给出的流语义以可能是无限长的动作序列（称为流）为域元素,而 C. A. R. Hoare 等人给出的失败语义则以进程史（*s*. *X*）为域元素,其中 *s* 是一个有限动作序列,称为进程史的踪迹,*X* 是一个有限动作集,称为进程史的拒绝集。第四类并发模型的语义有多种表示方法。这里列举三种。第一,变换语义学方法,采用形式化的手段把高层次的佩特里网变成低层次的佩特里网,以便用后者的语义描述前者的语义。但是这种方法不能解决低层次佩特里网的语义描述问题,因之是不彻底的。也可以把佩特里网变成其他的并发模型,然后再利用其他模型的语义来描述佩特里网的语义。第二,行为语义学方法,用佩特里网点火的记录来表示它的语义,其中的每个点火都可以是多个事件节点的并发点火,一个佩特里网的所有可能的有限点火序列构成一个佩特里语言。这种方法有点像进程代数模型中的流语义。第三,进程结构方法。按进程展开一个佩特里网,即得到一个类似于黎曼曲面的多层网。这种结构蕴含了该佩特里网的所有可能进程,可以作为佩特里网的语义。

每一种并发模型被提出时,通常都附有一种给定的语义,或者是一开始没有明确的语义,后来有人赋予它一种语义。例如,R. Milner 为 CCS 配备了交叠式语义,而 C. A. Petri 则为佩特里网规定了真并发语义。但这不等于说任何并发模型天生注定只能有一种语义。例如,利用多层佩特里网方法,同样可以为 CCS 建立一个完整的真并发语义。

并发模型的性质 作为一个形式系统,并发模型有许多重要的性质值得研究。这些性质大体上都是以进程为中心的。

通信：进程间交换数据称为通信,数据的发送和接收同时发生的称为同步通信,否则称为异步通信。

同步：各进程所执行的动作次序的互相制约称为同步。进程间的同步往往通过通信来实现。在佩特里网中用同步距离表示制约的严格程度。

死锁：多个进程为争夺同一资源（如设备使用权）而相持不下（谁也不能执行）称为死锁。

矛盾：多个进程为争夺同一资源而产生的互斥现象（只能有一个进程执行）称为矛盾。

活锁：由于资源使用权分配不当而使某个进程一直处于等待状态称为活锁。不存在活锁的系统称为是公平的。

活性：当前有动作可执行的进程称为是活动进程。任何时候都含有活动进程的系统称为是有活性的系统。

可观察：如果能从一个进程的外部测试出该进程执行了某个动作，则称此动作是可观察的。在 CCS 中，进程间的通信是不能从外部观察的。

发散：如果一个具有活性的系统所执行的任何动作都是不可观察的，则称此系统是发散的。

参考文献

1. Petri C A. Kommunikation mit Automaten. PhD Thesis, University Darmstadt, 1962

2. Milner R. A calculus of communicating systems. LNCS 92, Springer Verlag, 1980

3. Hoare C A R. Communicating sequential processes. Prentice-Hall, 1985

（陆汝钤）

bingxing bianyi chengxu

并行编译程序（parallelizing compiler） 处理并行语言的**编译程序**或串行语言的程序并行化的编译程序（自动并行编译程序）。

自动并行编译的实现在常规编译程序的基础上采取两种途径：预处理方式，在词法语法分析前进行；中间代码优化，在中间代码的基础上进行。预处理的方法，对数据依赖及循环中的数组下标分析清晰、精确，但并不适用于多种语言而导致重复劳动；中间代码的优化方法可通用化，但由于各语言实现时的中间代码形式各异且不对外开放，因而仅适用于系统制造者内部实现。自动并行编译的主要内容是并行性的识别。其过程是：数据依赖分析，即识别各种依赖如数据依赖、控制依赖等；程序转换，主要是循环的转换；借助运行系统和操作系统的支撑将源程序转化为并行代码或并行程序。自动并行编译一般进行源到源转换，即将串行源代码转换为并行源代码。

并行语言的编译程序通常由词法语法分析、优化和代码生成三个阶段组成。其中优化是主体，它包括依赖关系分析、循环转换及进程分配、调度、同步和通信等。

并行编译程序的主要研究内容包括：

（1）依赖关系分析

依赖关系主要有四种：流依赖、反依赖、输出依赖及控制依赖。其中只有流依赖是固有依赖，其他依赖如输出依赖和反依赖在一定条件下可通过换名扩张及设置必要的附加语句消去，而控制依赖亦可转化为数据依赖。循环的并行化主要是通过依赖关系分析，改变原串行程序的词法次序，以缩短程序的执行时间而不改变其语义。因此，依赖关系分析提供了保证串行程序语义正确性的条件。

编译程序依赖关系的测试方法主要有两种：精确测试法和近似测试法。精确测试法试图求出丢番图方程的通解以适合于方程数目较少且循环具有确定的上下界的情况，如依赖凸边域 DCH 方法。近似测试法仅给出一些存在解的必要条件，如 Banerjee 测试法、GCD 测试法及 λ 测试法等。

（2）循环转换

循环并行化之前应进行预优化工作，这包括循环的规范化（尽可能线性化以便于依赖关系的分析，使循环的上界和步长均为 1 等）。

循环的并行化可分为三个阶段：依赖分析（跨迭代和迭代内依赖）、循环转换和**循环调度**。循环的转换技术是对循环变量的集合进行划分或改变循环，主要技术有循环交换、合并、分布、分片和环路收缩等。循环调度则是为了取得更好的负载均衡，可分为静态和动态两类调度方法，这样的技术有：自调度 SS、块调度 CSS、引导调度 GSS 和梯形调度 TSS 等。

（3）过程分析

过程分析主要涉及别名分析、常数分析和引用分析，其目的是为了发掘被过程所隐蔽的信息以利依赖分析。对过程调用的处理，可在程序优化之前，采用内联扩展的办法即将被调用过程通过参数代换嵌入到调用处，最终使整个程序只包含唯一的过程，这种方法简单有效，但极耗内存，是一种消极的方法。已提出的过程分析技术有数组线性化、线性不等式、原子图及基于简单域的区域分析 RSD、RRSD 和 DAD 等。

自动并行编译工作最早始于 20 世纪 70 年代 CRAY-1 上的向量处理，当时只能识别某些适合于向量化的模式。70 年代末，美国 Illinois 大学研制了 Parafrase，在向量化和并行化领域进行了开拓性的研究。此后这方面的工作相继展开，及至 80 年代，

串行程序的自动并行化技术（围绕 FORTRAN）取得了长足的进步,其主要标志是数据依赖关系的分析技术的成熟、商品化工具（如 KAP 等）的出现及运用这些产品的系统开始获益。Illinois 大学的 U. Banerjee, M. J. Wolfe 和 Rice 大学的 K. Kennedy, R. Allen 等对数据依赖的分析和循环转换的研究,为向量化及后来的自动并行化工作奠定了理论基础。

由于现有程序中某些算法的固有顺序性,完全自动化的并行编译十分困难。事实上,一个性能良好的并行程序可能需要设计合适的并行算法,而不仅仅是并行性识别或代码转换。所以,增加并行语言成分和自动并行化的结合可能是今后发展的趋势。

参考文献

1. Banerjee U. Dependence analysis for super computing. Boston: Kluwer Academic Publishers, 1988

2. Ferrante J, Ottenstein K J, Warren J D. The program dependence graph and its use in optimization. ACM Trans. on Programming Languages and Systems, 1987, 9(3): 319-349

3. Allen R, Kennedy K, Porterfield C, et al. Conversion of control dependence to data dependence. In: Proc. of the Symposium on Principles of Programming Languages, 1983, 177-189

4. Wolfe M J. Optimizing supercompilers for supercomputers. Ph D. Thesis, University of Illinois at Urbana-Champaign, 1986

（谢立　朱根江）

bingxing chengxu sheji

并行程序设计（parallel programming） 利用并行程序编程语言编写运行于并行计算机上的并行程序。并行程序指若干操作同时执行,从并行化级别可以分为位级并行、指令级并行、数据并行以及任务并行。位级并行通常称为向量化并行,可以看为一种特殊的与硬件紧密相关的数据并行方式;指令级并行通常由编译器对指令静态排序和硬件动态调度配合实现,也可以通过手工优化汇编代码实现。而并行程序设计一般指面向不同问题设计并行算法,利用并行编程语言实现的过程,是一种面向问题的更为高层次的程序设计过程。

根据定义,可以从不同角度对并行程序设计进行分类。从并行计算机的软硬件数据共享角度,主要可以分为共享内存系统和分布式内存系统两种。在共享内存数据共享系统一般采用数据并行编程模型,例如 OpenMP 语言中对循环级代码的并行化,当然共享数据方式也必须同时具有同步、栅栏等操作的功能。在分布式存储系统上一般采用消息传递编程模型,例如利用 MPI 接口时,运行于不同处理器上的程序通过显式消息通信接口进行交互。另一种分布式共享内存系统为物理分离的内存提供虚拟的统一编址,例如 PGAS 语言族提供了相应的编译器和运行时系统,既提供了共享内存编程模型的高效编程效率,又保证了分布式系统的可扩展性。

从并行程序设计方式角度可以分为隐式并行设计和显式并行设计。隐式并行通常通过编程语言特定构建描述算法过程,例如 Chapel 语言中的域,不需要进行数据划分、计算同步、消息传输等设计,编译器和运行时系统自动处理并行化过程,但是隐式语言在提高编程效率的同时也会降低性能;显式并行方式中程序员负责任务划分、计算同步、数据通信等工作,编程效率低且容易出错,但追求性能的并行程序通常采用这种方式。

从并行程序语言通用性来讲,可以分为通用并行程序设计语言和领域并行程序设计语言。MPI、OpenMP、Pthread、OpenCL 等语言可以用来编写通用数据并行或者任务并行程序,而 MapReduce、Dryad、Pregel 等语言所提供的编程模型只适合某些特定领域内问题。

从并行程序执行角度,可以分为单程序多数据模型(single program multiple data, SPMD)和多程序多数据模型(multiple program multiple data, MPMD)。在 SPMD 中,所有处理器加载并运行同一可执行文件,每个处理器根据自己的唯一标识处理不同数据段或执行不同任务。在 MPMD 中,处理器运行不同可执行文件,通过显示消息传递或控制信号进行交互,MPMD 设计难度更大,并行程序设计通常采用 SPMD 模式。

类似串行程序设计模式,并行程序设计也存在一些通用设计模式,例如生产者消费者模式,线程池模式等。

（袁良　张云泉）

bingxing chengxu sheji yuyan

并行程序设计语言（parallel programming language） 程序员用来在并行体系结构计算机上进行程序设计的编程语言。为了帮助程序员解决并行程序设计面临的挑战性问题,并行程序设计语言的研究者们已经研究了许多种不同的模型和语

言,但迄今为止,还没有哪一种模型是通用的和完美的。按照访问存储器的方式,并行程序设计语言可以分为共享存储语言、消息传递语言和 PGAS 语言(partitioned global address space languages,分区全局地址空间语言)三大类。

在共享存储语言中,数据处在单一地址空间,分为共享和私有两种,数据通信通过共享存储来完成。平台独立的共享存储并行程序设计语言有 Pthread 和 OpenMP。POSIX Threads(简称 Pthreads)标准,由 IEEE 标准化委员会建立,在功能和接口方面都类似于 Solaris 的线程。当前最重要的共享存储标准是 OpenMP,它通过一组编译说明库例程和环境变量为 UNIX 和 Wondows NT 平台提供共享存储的应用程序接口。

在消息传递语言中,一个并行应用由一组进程组成,每个进程的代码是本地的,只能访问私有数据,进程之间通过传递消息实现数据共享和进程同步。消息传递的优点是用户可以对数据分布和通信实现完全控制,主要缺点是要求程序员显式地处理通信问题。在当前所有的消息传递语言中,最重要最流行的是消息传递接口标准 MPI(message passing interface)。北京大学并行软件小组开发的 Parray (parallel array)集群编程语言,根据源程序中统一表示的并行化信息自动产生低层并行代码;从访问存储器的方式来讲,也属于消息传递语言。

一大类语言是 PGAS 语言。在 PGAS 模型中,每一个处理器具有独立的存储器,但所有的处理器在逻辑上又被视为统一编址的全局存储空间;一个处理器上的进程可以访问全局存储空间中的任意地址,当该地址位于其他处理器的存储器时,将会引发处理器间的数据通信。PGAS 语言包括 Chapel、Co-Array FORTRAN(CAF)、HMPP、Hierarchically Tiled Arrays (HTA)、Titanium、Stanford PPL、UPC、X10、ZPL、Global Arrays 等。这些语言均从一定程度上支持对数组或域(domain)的操作,数据分布在不同处理器节点的方式有分块分布(block)、循环分布(cyclicity)等。CAF 通过在数组维度中引入称为 co-array 的维度来表示 SPMD 程序中的不同运行实例;Titanium 在 Java 中引入若干新特性以使得其适应高性能计算的需要:对多维数组、子数组的支持,为了在不同的 SPMD 程序实例之间进行协调,Titanium 支持一些同步和通信原语;ZPL 是由 Washington 大学开发的一种基于数组的并行语言,提供一系列的对于数组的操作模式,包括转置、广播、规约和 gathers/scatters 等,这些数组的操作符语法清晰,可以对应到程序执行相应的通信;UPC 通过对一个数组变量以 shared 关键字进行声明,该线性顺序的数组会以 cyclic 或 block-cyclic 的方式分布到不同的程序实例(或线程)上,并且 UPC 引入了一种新的循环结构 forall,类似于 **C 语言**中的 for 循环,但可以使用附加表达式将循环操作关联于不同的线程;在 HTA 中,数组数据被划分为所谓的分片(tiles),这些分片可以是传统的数组,也可以是嵌套的 HTA 数组,分片可以根据程序员指定的方式,分布在不同的处理器上,或者被存储在一个机器节点上。

参考文献

1. 安虹,陈国良. 并行程序设计模型和语言展. 软件学报,2002,13(1):118-124

2. Chen Yifeng, Cui Xiang, Mei Hong. PARRAY: a unifying array representation for heterogeneous parallelism. In: Proc of the 17th ACM SIGPLAN Symp on Principles and Practice of Parallel Programming, 2012

(陈一峯)

bingxing chuli xitong

并行处理系统(parallel processing system)

利用多个功能部件或多个处理机同时工作来提高系统性能或可靠性的**计算机系统**。

任何一个计算机系统都包含某种程度的并行性,但如果只具有硬件基本操作的并行性,如一个数据的所有位同时传送,许多门电路同时工作等,不能认为是并行处理系统。并行处理系统至少应包含指令级或指令级以上的并行。20 世纪 70 年代的流水线向量计算机在当时被认为是典型的并行处理系统,但后来用基于流水线技术的 RISC(参见**精简指令集计算机**)处理器构成的单机工作站,即使带不少外部设备和终端,一般也不认为是并行处理系统。所谓并行处理系统主要是指并行计算机系统或多处理机系统。

并行处理系统可以在 4 个级别上实现并行处理:指令内部、指令之间、任务或过程(程序段)之间和作业或程序之间。采用多个功能单元并行实现一条指令中的不同操作属于指令内部并行,**超长指令字**(VLIW)计算机(参见**多发射结构**)是实现指令内部并行的典型例子。同一时间执行两条以上指令称为指令间并行,超标量计算机(参见**多发射结构**)中有多条指令流水线,这是指令间并行的实例。一个程序往往可以分解成多个任务、子程序或过程,同一

程序内多个任务或过程可以在一个系统的不同处理机中同时运行,以缩短计算时间,称为任务级并行。多个作业或大型计算问题的多个独立的程序,在并行处理系统的不同的处理机或计算机中同时运行,以提高系统的吞吐量或有效地利用系统资源,称为作业级并行。

并行处理系统的研究与发展涉及计算理论、算法、计算机体系结构、硬件、软件(包括操作系统、编译、编程环境与程序语言等)以及性能评价等方面。并行处理系统与分布式处理系统有密切关系,随着数字通信技术的不断发展,两者的界限越来越模糊。从广义上讲,分布式处理(参见**分布式处理系统**)也可以认为是一种并行处理形式。

发展历程

20世纪40年代冯·诺依曼提出的细胞自动机是并行计算机的一种理论模型。1958年美国国家标准局研制的PILOT计算机由3台独立的处理机构成,可以简单地协同工作。在60年代和70年代,多功能单元计算机、阵列处理机和流水线向量计算机陆续从实验室走向市场,其中影响较大的并行计算机系统有CDC 6600,IBM 360/91,Illiac Ⅳ,TI ASC,CRAY-1等。在此期间也开展了多处理机研究,如美国卡耐基·梅隆大学研制的C.mmp和C*m等。80年代是向量计算机成熟的年代,以流水线技术为基础的向量计算机成为巨型计算机的主流技术。大规模集成电路的飞速发展与并行处理技术的逐步成熟为并行处理系统的发展提供了技术基础。就并行处理而言,90年代是多处理机时代,许多计算机公司推出共享存储多处理机系统或大规模并行处理机系统。并行处理的思想几乎与计算机同时诞生,但自电子计算机问世之后,经过半个世纪的努力还未真正普及,这一方面是由于器件水平的限制;另一方面是人们的思维方式受串行处理的束缚,并行软件发展较慢。

从历史上看,并行处理系统沿着几条不同的途径平行地发展。在共享存储的并行处理系统方面,共享存储系统的研制起始于20世纪70年代卡耐基·梅隆大学的C.mmp项目,采用交叉开关连接16个PDP 11/40和16个存储器模块。80年代纽约大学研制的Ultracomputer和伊利诺依大学研制的Cedar分别对多处理机的互联网、同步、并行编译以及性能评价等做出过重要贡献。90年代斯坦福大学的DASH和FLASH项目推动了分布式共享存储多处理机的研究。在消息传递并行处理系统方面,80年代初起始于加州理工学院的Cosmic Cube项目,后来发展成为Intel公司的iPSC并行机和Paragon大规模并行计算机。90年代初采用超立方体结构的Ncube并行计算机和麻省理工学院采用三维网格结构的J-machine实验系统对消息传递技术的发展也做出了贡献。在向量计算机方面,20世纪70年代初CDC公司推出了世界上第一台向量双机系统CDC 7600,后来向量计算机的体系结构分成寄存器到寄存器与存储器到存储器两个分支,Cray公司沿前一分支先后推出Cray-1,Cray-X/MP和Cray-Y/MP,成为巨型计算机的主要供应厂商,日本的Fujitsu, NEC, Hitachi等公司也先后推出过性能很高的向量处理机系统。CDC公司与ETA公司采用存储器到存储器结构,80年代曾推出过Cyber 205和ETA 10等向量机,到90年代这一分支已不再发展。在阵列处理机方面,20世纪60年代伊利诺依大学研制的Illiac Ⅳ,80年代Goodyear公司采用位片处理机构成的MPP和90年代初TMC公司的CM-2是阵列处理机的代表产品。在多线程并行处理系统方面,多线程并行处理的思想起源于CDC 6600的多功能单元,最早实现多线程处理的并行计算机是HEP,以后Tera并行计算机与麻省理工学院研制的Alewife计算机也发展了多线程技术。在数据流计算机(参见**数据流计算机**)方面,自从20世纪70年代初J. Dennis提出数据流计算机概念以后,数据流计算机的研究一直没有中断,除了麻省理工学院的Monsoon和*T项目以外,英国、日本也先后研制了Machester机、Sigma系列以及EM系列数据流计算机。

并行计算机分类

由于计算机技术十分复杂,很难用一种观点对并行计算机做精确的分类。较普遍采用的是M. J. Flynn于1966年提出的分类方法,即按指令流与数据流为1个或多个将计算机分为4类:单指令流单数据流(SISD)、单指令流多数据流(SIMD)、多指令流单数据流(MISD)和多指令流多数据流(MIMD)计算机。所谓指令流是指计算机执行的指令序列,数据流是由指令流所调用的数据序列。根据上述分类,除SISD计算机以外,其余3类都属于并行计算机。但是,很难设想一台计算机的多个指令流可以加工同一数据流,因此,多数人认为MISD计算机实际上不能实现,也有人认为将同一数据流以流水线方式进入由多个处理机构成的**脉动阵列**是MISD计算机。由于对数据流与指令流的理解不一样,一种

计算机可能划归不同的类型。例如，最初 Flynn 认为流水线计算机属于 SISD 计算机，后来他又将流水线计算机归类于 SIMD 计算机。

单指令流多数据流计算机　SIMD 计算机的一种类型是**阵列处理机**，它采用一个控制单元控制许多处理单元，每个处理单元同步地执行同一指令流。由于每个处理单元的数据相互独立，可采取数据并行方式工作。处理单元的数目可成千上万，甚至上百万个。阵列处理机适合做大型数组运算，专用性较强。Maspar 公司的 MP-1 和 MP-2 是商品化 SIMD 阵列处理机的代表。

向量计算机是另一类 SIMD 计算机。它采用 1 条或多条流水线，在标量处理机上附加很强的向量处理能力，特别适于做大型科学工程计算。一个向量中的各个元素都要进行同样的运算，一个运算（如加法或乘法）可以分成若干个步骤，将执行不同运算步骤的功能部件按先后次序连接形成一条流水线（参见**计算机流水线**），不等前一个向量元素运算完，下一个向量元素就可以进入流水线。流水线允许多个向量元素在同一时间间隔内执行同一种运算的不同步骤，即以时间上的重叠实现并行处理，从而提高计算速度。向量计算机往往采用很高的时钟频率，非常高速的器件和专门的散热技术，因而成本较高。Cray 公司的 Y-MP 和 C-90 是向量巨型计算机的代表。高性能的大型计算机系统也提供向量处理部件，如 VAX 9000 和 IBM 390/VF。

多指令流多数据流计算机　这种计算机由若干处理机或处理单元、存储模块和互联网组成，在每一时刻，不同的处理机执行不同的指令。MIMD 计算机包括具有单一地址空间的共享存储多处理机系统和具有多个地址空间的消息传递多计算机系统两大类。

共享存储多处理机系统　包含具有统一地址空间的共享存储器，各个处理机通过共享变量进行通信与同步，各处理机联系密切，实现高度的资源共享，因而又称为**紧密耦合并行处理系统**或紧耦合并行计算机。如果所有的处理机都有相同的权利访问外部设备，即每个处理机都可以运行操作系统、响应中断，没有主从之分，则称这种多处理机系统为**对称式多处理机**（SMP）系统。反之，如果只有 1 个主处理器执行操作系统，响应外设请求，其他的多个处理器只能运行用户程序，这种结构的多处理机系统称为主从式多处理机系统。

多处理机系统的共享存储器可以集中在一起，通过高速总线或交叉开关与各个处理机相连，这种结构的技术较成熟，实现较容易。集中式共享存储的主要缺点是扩展性差，难以构成大规模并行系统。共享存储器也可以分布在各个处理机中，构成分布式共享存储多处理机系统，这种系统既具有共享存储编程容易等优点，又有很强的扩展性，但实现较困难，特别是保持高速缓存的一致性需要专门的机制（参见**共享存储**）。

消息传递多计算机系统　是一种**松散耦合并行处理系统**，其中每一个计算机都有自己的地址空间和局部存储器，通过**消息传递**方式进行相互通信。也就是说，这类并行计算机采用互联网将许多联系较松散的计算机组成一个并行处理系统，其中每个计算机（常称为结点机）都运行自己的操作系统，以消息传递方式实现结点机之间的同步。这种并行机的各个结点机一般以插件方式插在相对集中的 1 个或几个机柜里，并具有集中的控制台。

消息传递多计算机系统适于运行非常大的并行程序，其通用性不如紧密耦合的多处理机系统。它的主要优点是可扩展性强，能连接几千台甚至更多台计算机以构成大规模并行处理系统，性能价格比高，其缺点是编程较困难，软件工具有待进一步开发。

工作站簇　亦称工作站机群或工作站网络。实际上是一种分布式的并行处理系统。通信技术的迅速发展使得有可能利用标准的高速通信网络将多台工作站或高档微型计算机互连起来，构成一个并行处理系统。这种系统的通信方式与消息传递多计算机系统类似，其区别在于前者利用电信网络通信技术，如**异步传送模式**等。研制这类并行处理系统的关键技术在于降低进程级通信的软件开销，即高效率的通信接口。

除上述分类之外，从用户使用的角度可将并行计算机粗略地分成细粒度、中粒度、粗粒度 3 大类。并行计算粒度表示并行处理系统中的处理机独立执行的基本程序段的大小，即平均执行多少条指令后必须与其他处理机通信及同步。细粒度并行计算机一般由大量较简单的处理机组成，大都是按同步方式工作的 SIMD 计算机。中、粗粒度并行计算机大都是按异步方式工作的 MIMD 计算机。并行计算粒度反映处理机的处理能力与通信能力的比值。一般，粒度越细，并行化程度越高，但相应的通信与同步开销也越大。共享存储多处理机系统常用于支持细粒度或中粒度并行，而消息传递多计算机系统则

多用于支持粗粒度或中粒度并行。

关键技术

并行处理与串行处理最主要的区别是并行工作的各个处理机或处理单元需要互相通信并实现必要的同步，以及一个任务需要分解并分配到各个处理机中协调地执行。因此，构造并行处理系统的关键技术包括处理机的互连、处理机之间的通信与同步以及处理机的调度策略。

互连网络是区分不同并行处理系统的重要特征。较普遍采用的互连网络包括多处理机**系统总线**、交叉开关、多级互连网络以及点到点的静态互连网络，如网格、超立方体以及胖树网络等。

存取非本地信息的通信延迟和保证并行计算结果正确性所付出的同步开销是影响并行计算机性能的主要因素。对于具有统一地址空间的**共享存储**系统，尤其是分布式共享存储系统，保持高速缓存一致性或设计适合并行的存取模式是最关键的技术。对于基于消息传递机制的松散耦合多计算机系统，消除结点机与互连网络接口上的通信瓶颈，实现通信与计算的重叠是提高性能的关键。实现并行处理系统的通信与同步需要在硬件支持与软件支持两方面折衷考虑，直接用硬件支持可获得较高性能，但实现较困难，成本高。采用软件实现时又要在系统软件与用户软件之间折衷选择，在用户层进行通信，不进入操作系统，开销会大大降低，但通用性差，也加重了用户负担。

并行计算机的效率在很大程度上取决于并行系统软件。并行计算机能否普遍推广的关键在于能否提供方便而有效的并行应用软件以及友好的人机界面。并行计算机的系统软件包括并行操作系统、并行语言、并行编译和并行编程环境。并行操作系统的一项重要任务是对多个处理机进行自动的任务调度，使互不相关的资源同时工作。操作系统核心的并行化可采用以下 3 种办法：主从式、浮动执行与多线程机制。多线程机制可以最大限度地开发核心代码的并行性。并行操作系统大都是在 UNIX 基础上进行扩充而形成的。

并行编程有多种模型，包括共享变量、消息传递、数据并行、面向对象、函数式与逻辑程序等。设计并行语言要么抛开现有的串行语言，要么在现有的串行语言上扩展并行功能。重新设计的优点是不受过去语言中串行思想的束缚，但与用户熟悉的语言不兼容。因此并行语言多采用扩充并行功能方式，保持与 FORTRAN 和 C 等语言的兼容。并行语言的编译也有几种不同做法，如编译制导或宏处理，对扩展部分做预编译处理以及自动化的并行编译等。用户最满意的编译器是能自动检测并行性并把串行程序转换成并行程序的优化重构编译，FORTRAN 语言的自动并行已有一些成果，C 语言也有一些并行库，但大部分并行程序还是靠程序员重写或人机结合半自动完成。除了语言和编译系统，并行编程环境还包括各种软件编程工具与支撑系统，其中最重要的是方便而有效的并行程序调试工具。编程环境是并行计算机与用户的界面，应使用户使用并行计算机像使用微型计算机、工作站那样方便。

应用与发展前景

由于信号在计算机中传播速度不能超过光速，而微电子工艺又受量子效应等物理规律的限制，半导体器件的开关速度与集成度已接近其物理极限，提高计算机系统的处理能力主要将依靠并行处理技术。并行处理已成为计算机领域中最受重视的研究领域之一。

并行计算机广泛用于大型科学、工程计算和大型事务处理。几乎所有的**工作站**与小型机厂家都生产了共享存储多处理机服务器，且已在金融、财贸、通信、交通、政府以及企事业单位管理等各个领域发挥了重要作用。并行计算机也已进入大学、科研单位和各个计算中心。其应用的典型实例包括数值天气预报、空气动力学计算（例如模拟风洞实验）、海洋动力学与天体物理计算、石油勘探、核反应模拟、基因分析与遗传工程研究、**模式识别**与**人工神经网络**计算、VLSI 芯片设计、化学反应速度预测以及有限元分析等。

巨型向量计算机的计算速度已达每秒几百亿次浮点运算。以微处理机为基础的大规模并行计算机比传统的向量计算机性能价格比高，将以更快的速度发展。大规模并行计算机中的处理机数量已达数千台甚至更多，计算速度已突破每秒万亿次。今后，大规模并行处理系统（包括具有数百万处理单元的 SIMD 计算机系统）将向每秒 10^{15} 次浮点运算的目标发展。商品化的共享存储多处理机已达到每秒几百亿次浮点运算的计算能力，性能上已达到**巨型计算机**的水平，但成本比传统的巨型计算机低一个数量级。由于共享存储多处理机系统通用性强，将会更广泛地普及。随着并行编译技术和并行编程环境的逐步成熟，具有分布式共享存储的 MIMD 计算机和以标准通信网为基础的工作站簇将成为并行计算机的主要产品。

参考文献

1. Bell G. Ultracomputers: a teraflops before its time. Communications of the ACM, 1992, 35(8): 27-47

2. Denning P J, Tichy W F. Highly parallel computation. Science, 1990, 250(11): 1217-1222

3. Hwang K. Advanced computer architecture: parallelism, scalability, programmability. New York: McGraw-Hill, 1993
（李国杰）

bingxing fangzhen

并行仿真(parallel simulation) 两个以上的独立处理机同时或并发地执行仿真程序,以达到高速完成同一仿真任务的过程与方法(参见**计算机仿真**)。

随着仿真研究的对象日益复杂,规模日益增大,尽管计算机技术在突飞猛进地发展,但仍然满足不了仿真应用的要求。人们分析过,单**中央处理器**的**数字计算机**的计算速度不可能超越电路信号传输速度的极限,即电场传播速度 3×10^8 m/s。如果传输线按 1 m 计算,则计算机的反应速度不可能超过 3×10^8次/s 操作。然而,三自由度空间飞行器的实时仿真要求大于 10^7次/s 操作,一个核反应堆的仿真要求大于 10^{10}次/s 操作,而大系统的优化研究则要求大于 10^{12}次/s 操作等。

在仿真等多种复杂应用的推动下,人们开发和研制了多种不同的并行处理技术和并行处理系统(参见**并行处理系统**)。尽管并行计算机系统有各种不同的结构,因而产生了各种不同的并行仿真处理技术,归纳起来,下述技术是所有并行仿真系统中的共性技术:

(1) 多机协调控制　多机并行协调的主要特征是并行运算和资源共享。随着处理机数目的增加,对资源的竞争以及由此产生的“死锁”和“排斥”问题日益严重。为解决这些问题,多机协调控制技术的有效性及完备性需要解决。目前,时序调度法、优先级调度法、超时自限法、开锁关锁法等方法在不同类型的并行仿真系统中得到应用。

(2) 机间同步及通信　多机并行完成同一任务,机间要频繁地交换数据,而且往往需要同步。高速地完成多处理机之间的有效通信,减少通信等待时间,确保必需的同步,一方面在硬件设计时就需要加以充分的考虑,在仿真算法及软件实现方面也要提供有力的支持。

(3) 并行仿真算法与并行仿真语言　为了适应多处理机并行计算的要求,必须有相应的并行仿真算法,并要有并行仿真语言加以实现。

(4) 并行操作系统　并行操作系统除了要完成系统资源管理外,还需进行任务分解分配、多机调度、同步及通信,并解决总线等资源竞争问题,还能支持并行仿真算法与并行仿真语言,从而为用户提供实现多机并行处理的环境。

参考文献

1. 肖田元,范文慧. 系统仿真导论. 2 版. 北京: 清华大学出版社,2010

2. 张晨曦. 连续系统仿真专用多处理机的探讨. 系统仿真,1987,(2)

3. 赵会平,潘刚,徐心和. 并行仿真技术综述. 计算机仿真,2003,20(11)
（肖田元）

bingxing gongcheng

并行工程(concurrent engineering, CE) 对产品和其相关的过程(包括生产过程和支持过程)进行集成化、一体化并行设计的一种系统方法。该方法可使产品开发人员从一开始就考虑到产品全寿命周期(从概念到报废)的所有因素,包括质量、成本、调度、用户需求、回收、环境影响等,从而减少产品设计早期的片面性和盲目性以及产品设计的循环修改次数,缩短产品设计周期。并行工程是一种先进的现代产品设计开发技术和工作模式。

20 世纪 80 年代,为了克服传统的串行产品设计开发模式的不足,并行工程概念被提出。1988 年,美国国防分析研究所提出了著名的《并行工程在武器系统采办中的作用》报告,明确提出了并行工程概念;美国在西弗吉尼亚大学组建了并行工程研究中心,推动了并行工程技术的早期发展。90 年代,发达国家针对并行工程实施了多项研究计划,包括美国的 DARPA/DICE 计划、欧洲的 ESPRIT Ⅱ & Ⅲ计划、日本的 IMS 计划等,有力促进了并行工程研究的深入开展。我国在 1992 年将并行工程列为国家高技术研究发展计划(亦称“863”计划)自动化领域计算机集成制造系统(CIMS)主题的重点研究内容,开展了一系列并行工程关键技术的研究与应用。从 20 世纪末至今,并行工程在继续进行理论研究的同时,开始了在工业界的广泛应用,并取得明显成效。

并行工程是一项内容十分丰富的综合技术和工作模式,总体上可分为技术和实施方法两个方面。

并行工程的主要技术内容包括：①面向并行工程的建模。包括面向并行工程的产品信息建模、产品开发过程建模、制造环境建模等；②面向并行工程的设计 DFX 技术。包括面向装配的设计(design for assembly,DFA)、面向制造的设计(design for manufacture,DFM)、面向产品结构的设计(design for product structure, DFPS)、面向工艺的设计(design for process, DFP)、面向维护的设计(design for service, DFS)、面向质量的设计(design for quality,DFQ)等；③面向并行工程的集成技术。包括支持并行产品开发的信息集成、功能集成、过程集成等；④协同技术。包括面向并行工程的计算机支持的协同工作(CSCW)环境、并行设计过程协调与控制、冲突检测与消解等；⑤面向并行工程的企业体系结构与组织机制。包括人的集成(客户、设计者、制造者和管理者)、企业各部门功能集成、信息集成及设计、制造工具集成的组织机制等；⑥并行工程管理方法。包括用户需求管理、产品服务管理等。

并行工程的实施方法主要包括：①创建并行的产品设计开发流程。使产品开发的所有活动尽早开始,使下游过程尽早参与到早期设计之中;在产品设计的早期阶段并行地考虑产品的可制造性、可装配性、可测试性、易维修性、成本、环境影响及市场销售等全部因素。②组建并行工程产品开发项目团队。产品开发团队由传统的专业分工部门制变成以产品为主线、跨各职能部门的多学科项目开发团队,从而加强部门间协调,集成多学科人员的智慧。根据产品复杂程度或项目大小的不同,可组成有不同级别层次的多个项目团队。③建立基于**产品数据管理**(PDM)/**产品生命周期管理**(PLM)的并行工程集成协同工作环境。建立支持资源分级别共享和协同工作的计算机网络平台,实现 CAX/DFX 等应用工具与其他管理工具的对接,以满足并行工程模式下的产品设计开发需要。

几十年来,并行工程技术日趋成熟,并在航空、航天、汽车、电子、机械等领域的很多企业得到了成功应用,取得很好的效果。20 世纪 90 年代,波音 777 客机的设计开发就是按照并行工程方法实现的。

随着工业界对并行工程技术的要求越来越高,并行工程技术自身一直在不断发展,其主要发展趋势为：①全生命周期化　在产品设计早期阶段能够有效并行考虑的产品全生命周期相关因素更加全面,甚至包含环境影响、节能等因素；②强功能　并行工程使能技术与系统的功能更强、健壮性更好；③集成化　异构系统的信息集成、功能集成、过程集成与互操作更加无缝；④协同化　网络化的协同工作平台支持全球化项目团队并行协同地开展产品设计开发；⑤规范化　并行工程实施方法更规范、更简单方便。

参考文献

1. 熊光楞,等. 并行工程的理论与实践. 北京：清华大学出版社, 2001

2. 宋贵宝,等. 武器系统工程. 北京:国防工业出版社, 2009

3. Withanage C, Park T, Choi H-J. A Concept evaluation method for strategic product design with concurrent consideration of future customer requirements. Concurrent Engineering: Research and Applications. 2010, 18(4): 275-289

(高曙明　何发智)

bingxing shujuku

并行数据库(parallel database)　以并行计算机为硬件环境并能充分发挥多处理和 I/O 并行性的**数据库系统**。

并行数据库技术起源于 20 世纪 70 年代的数据库机(database machine)研究。数据库机研究的内容主要集中在关系代数操作的并行化和实现关系操作的专用硬件设计上,希望通过硬件实现关系数据库操作的某些功能,该研究以失败而告终。20 世纪 80 年代后期,并行数据库技术的研究方向逐步转到了通用并行机方面,研究的重点是并行数据库的物理组织、操作算法、优化和调度策略。从 20 世纪 90 年代至今,随着处理器、存储、网络等相关基础技术的发展,并行数据库技术的研究上升到一个新的水平,研究的重点也转移到数据操作的时间并行性和空间并行性上。

并行数据库研究主要围绕**关系数据库**进行,包括以下 4 个方面：

(1) 实现数据库查询并行化的数据流方法　此种方法利用关系操作的固有并行性,可以较为方便的对查询做并行处理。此种方法简单、有效,目前已被很多并行数据库采用。

(2) 并行数据库的物理组织　此方法是研究如何把一个关系划分为多个子集合并将其分布到多个处理节点上去(称为数据库划分),其目的是使并行数据库能并行地读写多个磁盘进行查询处理,充分发挥系统的 I/O 并行性。**数据划分**对于并行数据库

的性能有很大影响，目前数据划分方法主要有三种，它们是一维数据划分、多维数据划分和传统物理存储结构的并行化。

(3) 新的并行数据操作算法　研究表明，使用并行数据操作算法以实现查询并行处理可以充分地发挥多处理机并行性，极大地提高系统查询处理的效率和能力。近年来许多并行算法已被提出，主要是围绕连接操作的算法较多，它们有基于嵌套循环的并行连接算法，基于 Sort-Merge 的并行连接算法以及并行 Hash-Join 算法。

(4) 查询优化　查询优化不仅是传统数据库的重要组成部分，也是并行数据库的重要组成部分。具有多个连接操作的复杂查询（简称 MJ 查询）的优化问题是查询优化的核心问题，并已取得不少可喜成果。

比较著名的并行数据库系统有 Arbre、Bubba、Gamma、Teradata 及 XPRS 等。随着并行计算机系统的发展，并行数据库也将会得到极大的发展。

参考文献

1. DeWitt D J, Gray J. Parallel database systems: the future of high performance database systems. Commun. ACM, 1992, 35(6): 85-98

2. 李建中. 并行关系数据库管理系统引论. 北京: 科学出版社, 1998

（李建中　徐洁磐　邹兆年）

bingxing suanfa

并行算法（parallel algorithm）　适用于并行计算的**算法**。所谓并行计算通常由一些可同时执行的进程描述来表示，这些进程在执行过程中相互作用和协调工作，以完成对给定问题的求解。例如，1000 个数相加的问题，将其分为 4 个进程，每个进程是一个部分和 $S_i(i=1,2,3,4)$，执行 250 个数相加。假设它们在由四台处理机 $P_i(i=1,2,3,4)$ 组成的并行机上进行计算，首先，每台处理机 P_i 执行一个进程，得到部分和 S_i，当四台处理机完成了它们的部分和计算后，其他三台处理机（如 P_2、P_3、P_4）向 P_1 发送部分和 $S_i(i=2,3,4)$，然后，P_1 将部分和 S_1，S_2，S_3，S_4 串行相加，从而完成 1000 个数的并行计算。如不考虑通信开销，该并行算法执行速度比串行算法执行速度约提高 4 倍。并行算法性能的好坏直接影响并行计算机的使用效率，它是提高并行计算机处理效率的主要因素之一。

并行算法随着 20 世纪 60 年代并行计算机的研制而兴起，并随着并行计算机的发展而迅速发展。它的发展经历了几个重要阶段：①公理化阶段。60 年代并行算法的研究建立在理想化并行计算机模型上，算法的复杂性度量建立在几条公理化假设上。②向量运算的并行算法设计阶段。70 年代**阵列处理机**和向量处理机（如 Cray-1，YH-1）相继研制成功并投入使用，并行算法研究进入实践阶段，研究内容主要基于流水线向量处理机的向量计算方法及其应用软件研制。70 年代末和 80 年代初是其研究的顶峰时期，著名成果有递归问题的向量化。③多向量处理机的并行算法设计阶段。它是随着多向量处理并行机（如 Cray Y-MP，YH-2）的出现而出现的，其特点是既要考虑多处理机间的任务级并行，又要考虑单处理机上的向量级细粒度并行。该类算法在 80 年代初期和中期比较流行。④多指令流多数据流（MIMD）类并行机上的并行算法设计阶段。该类并行机的显著特点是处理机功能比较强大，能独立处理各类复杂运算，处理机间通过显式或隐式的通信来相互协同，共同求解同一问题。该类并行机兴起于 80 年代初期，求解偏微分方程的区域分解算法是这方面的典型代表。⑤现代并行算法设计。目前，并行算法的研究主要集中于粗粒度 MIMD 并行算法，并要求具有可扩展性和易移植性，现代并行计算机的高性能获取要求并行算法设计兼顾两个方面：一是可扩展、易移植的粗粒度任务级并行；二是在每个进程，组织能充分发挥单机性能的合理数据结构、程序设计和通信方式。只有这样，才能真正发挥并行计算机潜在的性能。

1966 年 M. J. Flynn 提出一种并行计算机分类方法，为并行算法的研究提供了两种并行计算机模型，即单指令流多数据流计算机（SIMD）和多指令流多数据流计算机（MIMD）。以此为依据，并行算法可粗略地分为同步并行算法和异步并行算法两类，每一类又包含数值并行算法和非数值并行算法两种。**同步并行算法**是指算法各进程的执行中，一些进程的执行必须等待另一些进程的结果。**异步并行算法**是指算法各进程的执行均不需要相互等待，进程之间的通信是通过动态读、取存放于公用存储器中更新了的信息或随机地相互使用对方送来信息的一类算法。数值并行算法基本上属于数值分析范畴，即对以数学形式表示的问题求其数值解。非数值并行算法基本上属于符号处理的范畴，即对以字符、数字、图形或其他记号表示的问题进行并行处理。

并行算法的研究分为三个层次：

(1) 并行算法理论　主要研究不同并行计算模型的能力、限制和等价关系、计算问题的可并行性和下界技术等。计算模型实际上是硬件和软件之间的桥梁，使用它能设计与分析算法及算法的实现。目前流行的并行计算模型有 PRAM 模型、BSP 模型、LogP 模型等。PRAM 模型假定存在着一个容量无限大的共享存储器，有有限或无限多个功能相同的处理机，每个处理机具有简单的算术运算和逻辑判断功能，在任意时刻各处理机均可通过共享存储单元相互交换数据。BSP 模型将并行机的特性抽象为三个定量参数 p,g,L，分别对应于处理机数、带宽、全局同步之间的时间间隔。其计算由一系列全局同步分开的周期为 L 的超步组成，在各超步中，每个处理机执行局部计算，然后进行同步操作。LogP 模型将并行机的特性抽象为四个定量参数 L(延迟)、o(开销)、g(连续发送或接收消息的最小时间间隔)和 P(处理机数或存储器数)。其中 L、o 和 g 三个参数刻画通信网络的特性，但却屏蔽了网络拓扑、路由算法和通信协议等具体细节。

(2) 并行算法的设计与分析　并行算法设计有三种方法：一是检测和开拓现有串行算法中的固有并行性，直接将其并行化。这样，算法的稳定性和收敛性等复杂问题，在直接并行化后就无须考虑。二是修改已有的并行算法，使之可求解另一类相似问题，称为“借用法”。此类方法不但要从问题求解方法的相似性方面仔细观察，寻求问题解法的共同点，而且借来的方法要用得上，效率高，从而达到借用的目的。三是从问题本身的描述出发，设计一种全新的并行算法。通常包括平衡树法、倍增法、划分法、分治法、流水线法、加速级联和破对称方法等。

(3) 并行程序设计和性能优化技术　常用的**并行程序设计**有向量程序设计、共享存储并行程序设计和消息传递并行程序设计 3 种。**向量程序设计**以 1991 年 FORTRAN 90 语言的推出和许多自动向量化编译的研制成功为标志，已基本成熟。**共享存储并行程序设计**以宏任务、微任务和自动任务技术为代表。1990 年正式推出的 PCF FORTRAN 是共享存储并行程序设计的语言文本。**消息传递并行程序设计**是指用户必须通过显式发送和接收消息来实现处理机间的数据交换。每个并行进程均有自己独立的地址空间，相互之间的访问必须通过显式**消息传递**来实现，并行开销比较大，主要适合于粗粒度的并行计算。消息传递是当前并行计算领域的一个非常重要的并行程序设计方式，支持当前所有的分布式存储、**分布式共享存储**、**共享存储**和 NOWs/COWs 系统。

并行编程环境以 PVM、MPI、HPF、OpenMP 为代表。并行虚拟机(PVM)是比较流行的基于消息传递的并行通信库，异构性和易移植性是其设计的主要目的。消息传递界面(MPI)于 1994 年推出，目的是实现消息传递库的标准化，以保证消息传递并行程序设计的易移植性，已成为消息传递用户界面的工业标准。高性能 FORTRAN 在 FORTRAN 语言的基础上，提供一系列并行指导语句，用于描述并行程序中数据的存储方式和指导循环级粗粒度并行。OpenMP 于 1998 年推出，目的是实现共享存储并行程序设计的标准化，以保证共享存储并行程序的易移植性。目前，它在通常的程序设计语言的基础上，提供五类不同结构(并行区结构、工作共享结构、组合并行工作共享结构、同步结构和数据环境结构)的并行指导语句，以实现程序的循环级粗粒度并行。

程序性能优化技术主要有串行程序优化、共享存储环境下并行程序优化和消息传递并行环境下并行程序优化。串行程序优化技术有循环级优化、Cache 命中率优化、过程级优化、Branch 优化以及尽量采用标准库软件等。共享存储环境下并行程序优化技术有循环级优化、高速缓冲存储器一致性优化、粗粒度并行优化等。消息传递并行环境下的并行程序优化技术有减少通信开销、消除通信瓶颈和保持负载平衡等。负载平衡方法包括静态和动态负载平衡两种，其中静态是指并行模拟过程中，分配给各个处理机的任务是固定不变的，动态是指必须根据各处理机间负载平衡情况，动态调整各处理机的任务，以保证负载平衡。

并行算法的性能度量主要包括并行算法的运行时间、加速比与效率、并行算法的可扩展性等。并行算法的运行时间 t_p 是指求解一个问题的并行算法在 p 台处理机上从算法开始执行到算法执行完毕的这一段时间。**加速比**(又称绝对加速比)定义为 $s_p = t_s / t_p$，t_s 是指求解相应问题的最快串行算法在单处理机上的运行时间。效率 $E_p = s_p / p$，p 是并行算法所需的处理机台数。由于求解一个问题的最快串行算法很难找到，甚至不存在，因此目前采用相对加速比进行度量，它定义为 $s_p = t_l / t_p$，t_l 是指求解相应问题的并行算法在一台处理机上的运行时间。可扩展性是指随着并行系统规模的增大，通过适当增加问题的规模，使并行系统的性能与其规模成线性比

例增长。目前常用的有固定时间的加速比、等效率可扩展性度量、等速度可扩展性度量等。**固定时间的加速比**是指在保持问题规模与机器规模呈线性增长情况下度量加速比,如果加速比同机器规模趋于线性增长关系,则表明该系统是可扩的。**等效率可扩展性度量**是指系统规模增加时,测量增加多少运算量会保持效率不变。**等速度可扩展性度量**是指在保持系统的平均速度不变情况下考察问题规模与系统规模呈线性增长的能力。可扩展性不只依赖于机器规模和问题规模,也依赖于机器的存储器容量、I/O 能力以及通信能力等因素。

为了使读者对并行算法有个基本的了解,下面举两个例子加以说明。

(1) 矩阵转置 在多个处理机构成的 $n \times n$ 网格上对 $n \times n$ 的矩阵 $\boldsymbol{A}$ 求转置矩阵 $\boldsymbol{A}^{\mathrm{T}}$。设处理机 $P_{i,j}$存有矩阵元素 $a_{i,j}$,转置算法包含两个阶段,若将处于对角线上的处理机 $P_{i,i}$称为路由器,在第一阶段,每个对角线以上的处理机 $P_{i,j}(i<j)$将值 $a_{i,j}$往下以步进的方式传给处理机 $P_{j,j}$,到达 $P_{j,j}$需花费 $k=j-i$ 步。然后,$P_{j,j}$花费 $k=j-i$ 步将该值传送给 $P_{j,i}$处理机。第二阶段,所有对角线以下的处理机 $P_{i,j}(i>j)$将值 $a_{i,j}$向右以步近方式传给 $P_{i,i}$,然后 $P_{i,i}$将该值向上传送,以步进方式到达 $P_{j,i}$处理机。最终,每个处理机 $P_{i,j}$将含有值 $a_{j,i}$。转置算法可在 $O(n)$时间内完成,对于网格来说它是渐近最优的。

(2) 矩阵乘 设要计算两个 n 阶方阵 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 的乘积 $\boldsymbol{C}$。设 $p=m^2$(m 为正整数),将 $\boldsymbol{A}$,$\boldsymbol{B}$ 与 $\boldsymbol{C}$ 以适当方式分为 p 块 $\boldsymbol{A}_{i,j}$,$\boldsymbol{B}_{i,j}$和 $\boldsymbol{C}_{i,j}(0<i,j\leqslant m)$,每块为$(n/m)\times(n/m)$子矩阵,将这些块映射到 $m \times m$ 个逻辑处理机阵列上,每个处理机记为 $P_{i,j}(0<i,j\leqslant m)$,$P_{i,j}$存储 $\boldsymbol{A}_{i,j}$,$\boldsymbol{B}_{i,j}$,并计算 $\boldsymbol{C}_{i,j}$。在计算 $\boldsymbol{C}_{i,j}$时,需要所有子矩阵块 $\boldsymbol{A}_{i,k}$,$\boldsymbol{B}_{k,j}(0<k\leqslant m)$,因此 $\boldsymbol{A}$ 的块在处理机阵列每行作多对多广播,而 $\boldsymbol{B}$ 的块在处理机阵列每列作多对多广播。当 $P_{i,j}$接收完 $\boldsymbol{A}_{i,1}$,$\boldsymbol{A}_{i,2},\cdots,\boldsymbol{A}_{i,m}$;$\boldsymbol{B}_{1,j},\boldsymbol{B}_{2,j},\cdots,\boldsymbol{B}_{m,j}$后,执行子矩阵乘法和加法运算。在每台处理机上,计算时间为 $O(n^3/p)$,而对应的串行时间为 $O(n^3)$。

参考文献

1. 李晓梅,莫则尧,等. 可扩展并行算法的设计与分析. 北京:国防工业出版社,2000

2. 陈国良. 并行计算——结构、算法、编程. 北京:高等教育出版社,1999

3. Dongarra J, Foster I, Fox G, et al. Sourcebook of parallel computing. Burlington, MA: Morgan Kaufmann Publishers, USA, 2003

(李晓梅 莫则尧 陈军)

bingxing xuniji

并行虚拟机(parallel virtual machine, PVM) 一种面向异构集群环境的分布式编程模型,其目的是将网络环境中异构计算机抽象为虚拟机,使之支持并行处理。PVM 支持 C、C ++ 和 FORTRAN 语言,提供 master/slave、单程序多数据(single program multi-data)等多种编程模式。PVM 还支持通信缓冲区的动态管理机制,每个消息所需的缓冲区由 PVM 运行时动态申请,消息长度只受结点上可用存储空间的限制。PVM 提出了进程组的概念,可以把一些进程组成一个进程组,一个进程可属于多个进程组,而且可以在执行时动态改变,并提供一组消息通信原语支持进程间通信,包括广播和多播等操作。

PVM 最早由美国的田纳西大学、橡树岭国家实验室以及埃默里大学于 1989 年发布,后来由田纳西大学重写,并于 1991 年发布了版本二,版本三于 1993 年发布,支持容错,即当发现一个结点出故障时,PVM 会自动将之从虚拟机中删除。相比 MPI,PVM 更适合于异构平台,支持更丰富的编程模式、容错以及动态资源和进程管理。MPI 相比 PVM 提供了更多消息传递模式和函数,包括更丰富集合通信算法,支持通信拓扑管理,并且具有更高通信性能。

参考文献

Dongarra D, et al. PVM: Parallel Virtual Machine: A Users' Guide and Tutorial for Network Parallel Computing. MPI Press, 1994

(袁良 张云泉)

bofen fuyong

波分复用(wavelength division multiplexing, WDM) 将不同波长的信号复用到一根光纤上传输的技术。波长和频率呈反比关系,所以,波分复用技术并不是一个严格意义上的新技术,而只是**频分复用**技术的一个变种。

波分复用技术的工作原理如下:

(1) 输入端的每路光信号处于不同的波长处,组合器将其组合成一个光信号,并输出到复用的光纤上;

(2) 通过复用的光纤,复用的光信号被传输到远方的接收端;

(3) 在接收端,复合光束被分离器分离出和输

入端一样多的不同波长的光信号。

在图1所示的波分复用原理图中，需要注意的是：

(1) 4个光信号可以推广到 N 个，N 值的大小受制于组合器和分离器的处理能力以及光纤的复用能力；

(2) 波分复用光纤通信系统分为单向和双向两类，双向系统需要使用双向耦合器来实现波分复用；

(3) 组合器、分离器、双向耦合器都可以使用WDM复用器来称呼，它是波分复用光纤通信系统的关键部件；

(4) 相对于FDM来说，光纤通信系统通常采用无源的衍射光栅，是完全被动的，因而也是非常可靠的；

(5) 以前，复用的信道每100 km就需要进行光电转换，放大之后，再转回光信号，继续传输，以便长距离传输；现在，有了全光放大器，它可以重新产生整个信号，单独放大，可每1000 km做一次，无须进行多次光电转换，大大地提升了干线的传输性能。

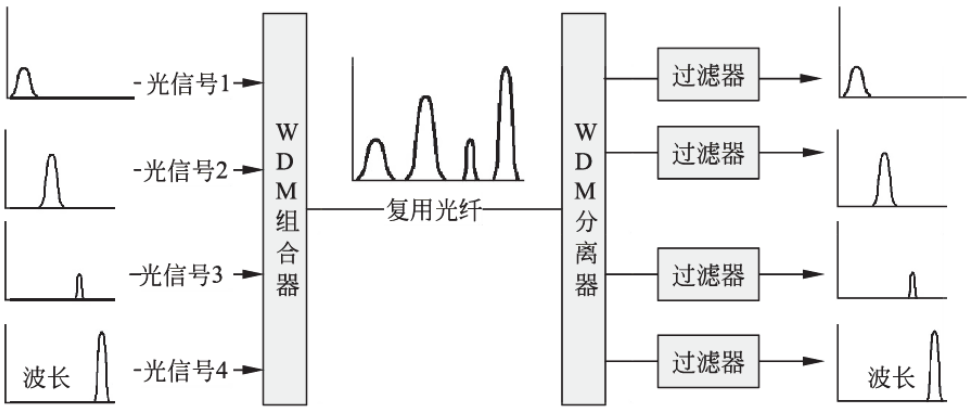


图1　波分复用原理示意图

WDM技术发明于20世纪90年代。第一个商业WDM系统有8条信道，每条信道的带宽为2.5 Gbps；1998年，出现了40条信道的商业WDM系统，每条信道带宽是2.5 Gbps；到了2001年，市场上有了96条信道的产品，且每条信道的带宽为10 Gbps，这样总共的带宽可达960 Gbps。实验室还在研制具有更多信道的波分复用系统，当信道的数量很多，波长的间隔将变小(比如0.1 nm)，这时候的WDM通常称为密集波分复用(dense wavelength division multiplexing，DWDM)。

WDM技术得到了广泛的使用，发展也非常迅速。单根光纤的带宽约为25 000 GHz，从理论上计算，即使每赫兹传输1位信息，仍然有2500条10 Gbps信道的发展前景。

参考文献

1. Tanenbaum A S, Wetherall D J. 计算机网络. 北京：清华大学出版社，2012

2. 纪越峰. 现代通信技术. 北京：北京邮电大学出版社，2010

3. 张辉，曹丽娜. 现代通信原理与技术. 西安：西安电子科技大学出版社，2008

(袁华)

bofeng han

波峰焊(wave-soldering)　利用熔融的焊锡波，将预先插上的所有元器件一次焊接到印制板上的一种工艺过程。波峰焊焊接方式一般采用机械泵或电磁泵，使熔融的铅锡合金焊料形成一种持续的焊锡波。焊接时将插好元器件的印制板装在传送框架上或搬送爪上，以恒定的速度连续地输送印制板，使印制板下面的焊盘全都接触到焊料，从而完成焊接的全过程。

波峰焊在焊通孔元件时，焊料润湿元器件引线突出的部分，由于毛细孔作用，焊料被吸入印制板的金属化孔里。平面安装型(SMD)元器件的焊接，则是将元器件先粘在印制板的焊接面，直接浸入焊料的波峰中。

整个波峰焊包括三道工序：助焊剂涂布、印制板预热、波峰焊接，参见图1。

助焊剂涂布装置　助焊剂涂布装置用于在印制板上涂布助焊剂，其作用是除去锡波及焊接点的表面氧化物，使焊锡能在金属表面展开，降低溶锡表面张力，从而在元器件引线和印制板焊盘间形成良好的焊点。

(1) 助焊剂涂布主要采用以下两种方式。①发泡式　发泡式是以发泡石配合空压气体，形成发泡

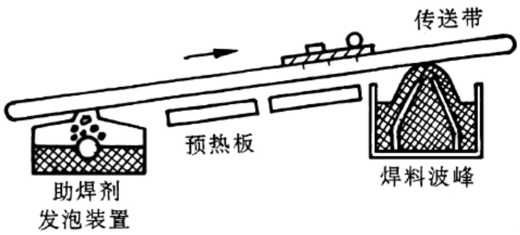


图 1　波峰焊系统示意图

的助焊剂，当发泡助焊剂通过喷嘴喷射到印制板焊接面，气泡立即破裂，在板面沉积上一层薄而均匀的助焊剂层，多余的助焊剂则流回到焊剂槽里。板上助焊剂涂布厚度决定于焊剂的黏度，先进的波峰焊机均有焊剂比重自动监控装置，以控制焊剂的黏度。②喷雾式　有采用压缩空气和采用超声波两种方法。前者是在焊剂槽中放置一个旋转的滚筒，滚筒周围装有细孔金属网，压缩空气通入滚筒并从金属网的细孔中喷出，带动焊剂形成喷雾。后者是用超声振子使助焊剂雾化。

(2) 印制板预热　其装置常见的波峰焊机预热装置有电热管、石英管、电热板等方式，一般预热到 80～120 ℃。对预热装置的要求是能在短时间内将印制板预热到所需要的温度，又能防止过度的热冲击；同时要使助焊剂汽化以便于清洗和加速助焊剂的化学反应，从而提高助焊性能。

(3) 波峰焊接　波峰焊接采用熔融的焊料。焊料由泵加压后经过一个通道，同时溢流到延伸板上，再回到焊料槽完成印制板的焊接。波峰焊设备主要有单波峰和双波峰两类。单波峰焊料与印制板仅有一个接触面，焊料波峰宽而且平稳；而双波峰焊料与印制板有两个接触面，两个接触面的大小和角度均不相同，以避免局部虚焊。

为适应平面安装型元器件的波峰焊的要求，已设计出多种新的锡波波形，以便在焊接过程中消除助焊剂和黏胶产生的气泡，消除一般稳定层流锡波易发生的阴影效应，避免因焊接时间太长使元器件因过热而损坏。新的波形有双波峰式、振荡式、气泡式、喷射式等。

波峰焊比手工焊一致性好，可靠性高，但不适用于塑封有引线芯片（PLCC）以及间距 0.65 mm 以下的四列扁平封装（QFP）。　（顾本斗）

Bosite duiying wenti

波斯特对应问题（Post's correspondence problem）　由 E. Post 于 1946 年提出的一个**不可判定问题**。

设 A 是非空字符有限集，A 中字符的有限序列称为 A 上的字符串。A 上的 m 个字符串 $u_1,\cdots,u_m$ 的连接字符串，记为 $u_1\cdots u_m$。A 上的表是 A 上字符串的有限序列，序列的长度称为表的长度。例如，$ab,\wedge,aa$ 是字符集 $\{a,b\}$ 上的长度为 3 的表，其中 $\wedge$ 表示空串。

设 $l_1,\cdots,l_k$ 和 $m_1,\cdots,m_k$ 是 A 上两个表，若存在小于或等于 k 的正整数 $i_1,i_2,\cdots,i_n$ 使 $l_{i_1}l_{i_2}\cdots l_{i_n}=m_{i_1}m_{i_2}\cdots m_{i_n}$，则称表 $l_1,l_2,\cdots,l_k$ 和 $m_1,\cdots,m_k$ 有匹配。例如，$A=\{0,1\}$，$l_1=1$，$l_2=10111$，$l_3=10$，$m_1=111$，$m_2=10$，$m_3=0$，因 $l_2l_1l_1l_3=m_2m_1m_1m_3=101111110$，因此，$l_1,l_2,l_3$ 和 m_1,m_2,m_3 有匹配。判定同一字符集上的任意两个相同长度的表有没有匹配的问题，称为波斯特对应问题。

由于**图灵机**的停机问题是不可判定的，可以推出波斯特对应问题也是不可判定的，即不可能找到一个算法来判定同一字符集上的任意两个相同长度的表是否有匹配。

波斯特对应问题在形式语言理论和程序设计理论中有重要应用。由于波斯特对应问题是不可判定的，可推出形式语言理论和程序设计理论中的许多问题是不可判定的。

参考文献

Hopcroft J E，Ullman J D. 自动机理论、语言和计算导引. 徐美瑞，译. 北京：科学出版社，1986

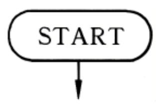
（陈火旺　贲可荣）

Bosite ji

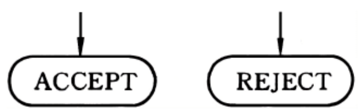
波斯特机（Post machine）　E. Post 提出的一种由表示宏操作指令的语句组成的计算模型。

在字母表 $\Sigma=\{a,b\}$ 上的一台波斯特机可以用一个带取值于 $\{a,b,\#\}^*$ 的字变量 x 的流程图来刻画。该流程图由下述语句组成：

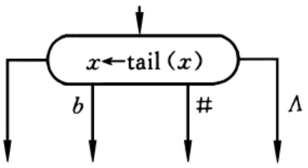
(1) 启动语句



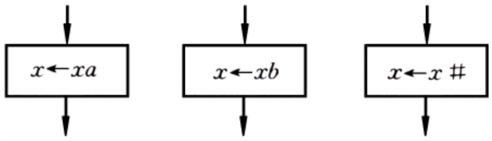
(2) 停机语句



(3) 分支判断语句



(4) 赋值语句



其中,Λ 表示空字(长为 0 的字),tail(x)表示将字 x 的最左一个字母去掉后所获之字,xa 表示将字 x 右接上字母 a 后所获之字。分支判断语句将字 tail(x)赋给 x,然后依据未赋值前的字 x 的最左字母而流向下一语句。在字母表 Σ 下的一个字 α 被称为可被波斯特机 M 接受(拒绝)的,若该波斯特机在输入 $x=\alpha$ 时计算最终流到停机语句 ACCEPT (REJECT)。E. Post 证明了:在字母表 Σ 下的波斯特机器类与图灵机器类识别语言的能力是相同的。波斯特机作为计算数论函数来说,可计算一切图灵可计算函数,而且只需使用 $x:=x+1$,$x:=x-1$ 形的赋值语句和 GOTO n, IF $x\neq0$ 形的分支判断语句。

(李祥)

boyishu sousuo

博弈树搜索(game tree search) 一种具有完备信息特征的双人博弈问题求解方法。在这类博弈问题求解过程中,两位选手对弈,双方轮流走步,每一方不仅完全知道对方过去已走过的棋步,而且还能估计出对方未来可能的弈着。对弈的结果是一方赢(另一方输),或为和局。这类博弈的实例有:一字棋、西洋跳棋、中国象棋、围棋等。而带机遇性的博弈问题,如掷骰子和大多数扑克牌游戏,因不具有完备信息特征,所以都不属于这里讨论的范围。

博弈问题可以用产生式系统模型来描述。例如对国际象棋,综合数据库可用以描述棋盘上棋子位置的布局,产生式规则可用以描述各种棋子的合法走步。这些规则作用于数据库,产生出所谓的博弈图或博弈树。

博弈树是一类特殊的"与或"树。树的根节点代表博弈的初始状态,它的后继节点是对弈的第一位选手相对于初始状态可能走出的所有棋局。而它们的后继节点又是第二位选手相应可以走出的所有棋局,以此类推。博弈树的叶节点表示棋局的输、赢(一方的赢标志另一方的输)及和局。例如,设 MAX,MIN 二人对弈,MAX 先行。对于任意给定的当前棋局节点,MAX 可能有几种走法,即可以生成当前节点的若干后继节点,这些后继节点被称为"或"节点。对于 MAX 可能走出的每一个棋局,MIN 可能分别有若干种走法应付。对 MIN 的所有这些走法,MAX 必须有所准备。因此,MIN 的后继节点被视为 MAX 的"与"节点。在此每个"与"节点之后又可继续下接 MAX 可能生成的数个"或"节点。"与"、"或"节点如此交替出现,直至达到标志胜、负、和局的叶节点,则一棵博弈树生成完毕。

在博弈树中搜寻使某方取胜的策略,类似于在"与或"图中搜索一个解图。例如,若要使 MAX 取胜,那么在对博弈树从开局到终局的各层节点的搜索过程中,只需保证 MAX 对一个"或"节点取胜,但必须保证它对所有有关的"与"节点取胜。这是完全取胜的策略。实现这种策略就是在博弈树中搜索一个解图,解图代表了一种完整的博弈策略。这种寻找完整博弈策略的做法只适应于求解简单的博弈问题(如 Grundy 问题求解等),或收拾复杂博弈的残局。对于复杂的博弈问题,如中国象棋等,由于受时间和存储空间的限制,这种博弈搜索策略是不适用的,这时就需要采用有界深度极小极大博弈策略。

有界深度极小极大博弈策略的思想是:每当选手走步之前,可根据当前的棋局势态,预测出双方对弈的若干步数,并以此构造出一棵有界深度的局部博弈树,树的深度亦为所预测到的步数。然后,根据极小极大原则标记树中各棋局节点的势态优劣,并从可能的走步中选择一步相对最好的棋着。这种策略又称为"一步好棋"策略。策略中的极小极大原则是指:对手永远不会犯错误,他总是走出对自己最不利的棋着,自己则应从这种给定最坏的棋局中选择相对最好的棋着。复杂博弈问题的求解过程就是不断地重复执行上述构造局部博弈树,选择最好棋步,直至博弈结束的过程。

为了能够判断局部博弈树中各棋局节点的好坏,通常需要定义一个静态估价函数 $f(P)$,用来估计棋局 P 的势态优劣。这个静态估价函数 $f(P)$ 可以定义为棋局 P 到我方胜利棋局的距离(如,胜利在望时:$f(P)\to+\infty$;趋向败局时:$f(P)\to-\infty$;势均力敌时:$f(P)=0$)。$f(P)$ 也可定义为棋局 P 到

某个明显有利于我方棋局的距离，如吃掉对方一子，等等。

在生成一棵局部博弈树之后，可用静态估价函数 $f(P)$ 去度量该博弈树中各个端节点 P 的价值，而后采用倒推的办法，根据极小极大原则，自下而上地计算博弈树中各个非端节点的估价函数值。图 1 所示就是从 MAX 的立场出发构造的一棵深度为 2 的局部博弈树。树中各节点的估价函数值 $f(P)$ 的计算过程如下：博弈树中各端节点给出的数字是根据原定义的静态估价函数 $f(P)$ 计算所得，而其他非端节点的估价函数值则是应用倒推的办法，按极小极大原则求得的。

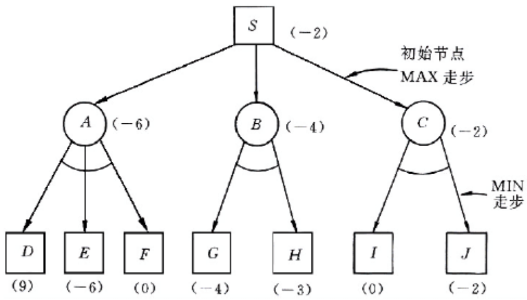


图 1　深度为 2 的局部博弈树 $f(P)$ 求值过程

总之，有界深度极小极大博弈搜索过程分为两个阶段：第一阶段用宽度优先搜索方法生成规定深度的局部博弈树，然后计算树中各端节点的静态估价函数值；第二阶段根据极小极大原则自底向上逐级求树中各非端节点的倒推估价值，直至求至初始棋局节点 S 的 $f(S)$ 值为止。

选手可由 $f(S)$ 值选得相对最好的走步，搜索过程暂告结束。对手响应走步之后，再以当前棋局状态作为起始状态 S，重复调用上述过程。

参考文献

1. Nilsson N J. 人工智能原理. 石纯一，等译. 北京：科学出版社，1984

2. 林尧瑞，马少平. 人工智能导论. 北京：清华大学出版社，1989

3. 陆汝钤. 人工智能. 北京：科学出版社，1989

（康建初　杨莉）

bujianduan dianyuan

不间断电源（uninterruptible power system，UPS）　当交流输入电源的变化（包括电压变化、频率变化及波形失真等）超出规定范围时，仍能正常地继续向负载输送能量的供电设备。计算机工作时不间断电源（UPS）可保证供电不间断。UPS 分为旋转发电机式和静止变换式两大类，后来派生出动静结合式（不间断电池系统）。不间断电源主要由换能、储能和传输等部分构成。

不间断电源的种类及其结构

1. 旋转发电机式 UPS　一种利用发电机来达到不间断供电目的的设备。它有三种：带飞轮的交流电动机-交流发电机组，带电池组的直流电动机-交流发电机组和内燃机 UPS 系统。由于设备笨重现在已很少使用。

2. 静止变换式 UPS　利用半导体器件、电池等组成的系统来实现不间断供电的设备。其基本结构如图 1 所示。

（1）输入滤波器　主要的作用是抑制天电干扰、工业干扰及其他电磁干扰、浪涌等外来干扰，以保护 UPS 内部元器件和电路的安全。

（2）整流器、充电器　将输入交流电压变成直流电压的装置。整流器一般采用晶闸管整流电路，以达到稳压目的。

（3）电池组　起储能作用。一般采用密封式免维护铅酸蓄电池，也有的采用碱性蓄电池，如铬镍电池等。

（4）逆变器　将直流电压转换成交流电压的装置。

（5）隔离变压器　将逆变器送来的交流电压变换成负载所需的值。

（6）转换开关及其监控器　转换开关在其监控器的控制下，根据 UPS 的运行状态进行转换。

静止变换式 UPS 按工作模式可分为在线式、后备式及在线后备混合式，按其结构又可分为双变换式和线路交叉式。

（1）后备双变换式 UPS（参见下页图 1）　在电网正常供电时，UPS 的输出电压由工作旁路支路提供，只有当电网异常时，其转换开关才打向下方，输出电压才由电池组—逆变器—隔离变压器这一支路供给。在电网正常时，电网输入除一路向输出供电外，另一路经输入滤波器、整流器、充电器支路向电池组充电。后备式 UPS 的优点是效率高、可靠性高、结构简单；缺点是供电质量差，有切换时间，输出波形一般为方波或梯形波。

（2）在线双变换式 UPS（参见图 1）　无论市电正常与否，UPS 输出电压始终取自输入滤波器—整流器、充电器—逆变器—隔离变压器支路，这时转换开关的触点与下端（隔离变压器）相合。当逆变器

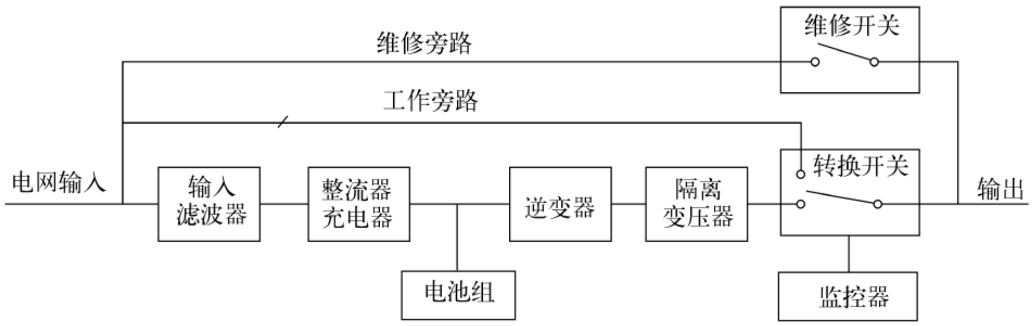


图 1　静止变换式(双变换式)UPS 基本结构图

故障或过载时,开关才打向上方,改由工作旁路供电。在线式 UPS 优点是供电质量高,市电断电时无切换时间(因多为静态开关),大都输出正弦波。缺点是效率低,结构复杂。

(3) 在线线路交叉式 UPS　基本结构见图 2。线路交叉式与双变换式不同的是在旁路上加进了滤波器、电网电压调节器等装置。在电网正常时,两条支路同时工作或交叉工作。当电网出现故障时,由下支路单独供电。线路交叉式具有高效率、结构简单、供电质量高和切换时间为零的优点。

(4) 在线后备混合式 UPS　结构原理见图 3。它由两条支路组成。当电网正常时,电网输入经过整流器送逆变器,同时电网给电池组充电,如图中实线所示。当电网异常时,则由电池组通过直流变换器向逆变器供电,如图中虚线所示。由于采用了综合器,没有转换开关,切换时间为零。

3. 动静结合式 UPS　由一个静止变换式 UPS 和一个直流发电机组成。当市电正常时,UPS 正常运转;一旦市电异常,UPS 由机内电池供电;如果市电短时间不能恢复,当电池放电到一定电平时,自动启动直流发电机为 UPS 机内电池充电,使电池的放电不间断地维持下去。

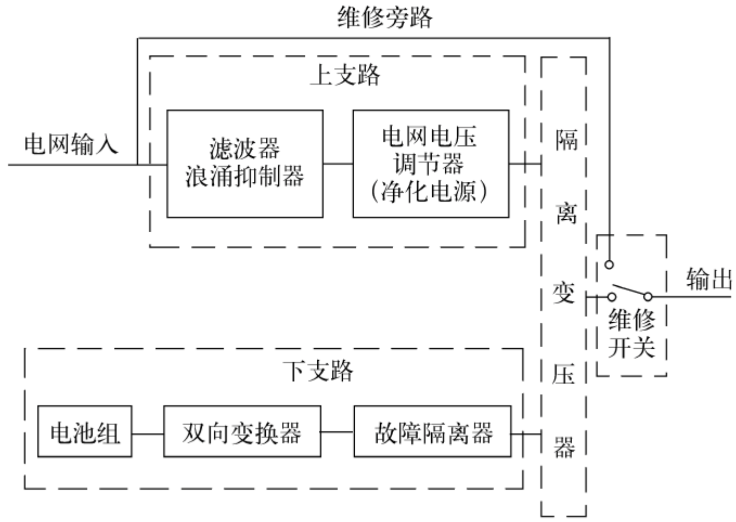


图 2　线路交叉式 UPS 基本结构

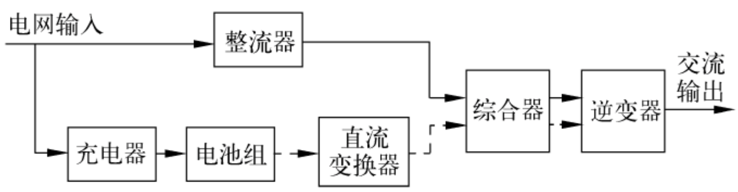


图 3　在线后备混合式 UPS 结构原理

不间断电源的技术规格

UPS 的主要技术指标如下：

（1）容量　指 UPS 的输出功率，其单位通常用 VA 表示。

（2）输入电压范围　指 UPS 不用电池供电时，电网输入电压的最大值和最小值。

（3）负载稳定度　当负载在指定范围变化时，UPS 的输出电压变化范围。

（4）效率　UPS 输出功率与输入功率之比。

（5）输出电压谐波失真　指 UPS 输出电压中高频分量所占比例。

（6）音频噪声　UPS 工作时机内发出的可听见的声音。

（7）切换时间　将负载从电网供电支路转换到电池供电支路或相反过程中，由于转换开关的作用，负载上会出现无电的一段时间，这段时间就是切换时间。

UPS 的应用与发展

广泛应用的 UPS 是静止变换式 UPS，它大量用于各种微型计算机供电系统中。由于微型计算机内采用的是开关电源，对输入波形要求不严格，正弦波、准方波和梯形波都能适应，因此常用后备双变换式 UPS。

对计算机联网系统，既要求供电质量高，又要求可靠性高，因此多采用在线式 UPS。为了提高可靠性，还要采取两台以上 UPS 并联、串联等冗余措施。先进的 UPS 内部采用微型计算机监控，通过 RS-232C 接口与外部通信，已成为计算机系统的一个有机部分。

除了在计算机领域外，UPS 在通信等领域也有广泛的应用。

参考文献

1. 中国电源学会交流稳定电源专业委员会. 第二届 UPS 技术研讨会论文集. 1994

2. 张立，赵永健. 现代电子电力技术. 北京：科学出版社，1992

（王其英　何春华）

buke panding wenti

不可判定问题（undecidable problem）　没有求解算法的问题（参见**可计算性理论**）。集合的从属关系所对应的判定问题，若不是可判定的，则称为不可判定的。数学和计算机科学中提出的大量判定问题中，其绝大部分都是不可判定的。研究不可判定性，对一些具体的判定问题给出否定回答，常常具有重要意义。因为一旦某个判定问题被证明是不可判定的，则表明解决这个问题的能行办法是不存在的，不必再去寻找解决该问题的能行办法。

研究不可判定性的两个基本方法是直接方法和间接方法，直接方法是使用康托对角线化方法；间接方法就是将已知的不可判定问题归约到所研究的判定问题上，从而证明该判定问题比已知的不可判定问题更不可判定。利用归约不但可以间接地证明某些判定问题是不可判定的，更重要的是归约关系在各种判定问题之间建立了不可判定程度的比较，依此确立了不可判定性的系统理论。

参考文献

1. Davis M D, Weyuker E J. Computability, complexity and languages, fundamentals of theoretical computer science. Academic Press Inc., 1983

2. 莫绍揆. 递归论. 北京：科学出版社，1987

（马绍汉　李大兴）

buquedingxing tuili

不确定性推理（uncertainty reasoning）　采用形式化方法表示和推理知识系统中的不确定性。

知识系统中的不确定性是指命题的真实性不能完全肯定，而只能对其为真的程度给出某种估计。例如，阴天打雷，可能下雨，也可能不下雨。因此，“阴天打雷会下雨”的命题具有不确定性。不确定性推理包括表示和推理两部分。不确定性表示是描述知识和证据的不确定性的形式化方法。不确定性推理是从已知不确定证据推出结论为真程度（或当新证据出现时，修正已推出结论为真程度）的过程。

不确定性一词最早出现于 1836 年 James Mill 的“政治经济学是否有用”一文中，后来在众多领域被广泛应用。1977 年，E. A. 费根鲍姆（Edward A. Feigenbaum）提出了**知识工程**的概念。知识工程主要研究如何利用人类专家的知识解决现实世界问题。由于客观世界的复杂性、多变性和人类自身认识的局限性和主观性，致使人类专家的知识和解决问题所使用的数据中往往含有随机、模糊、不完全和不一致等多种不确定性。知识系统要想达到人类专家水平，必须处理好知识和数据的不确定性问题。因此，不确定性推理成为人工智能领域的一个重要研究方向，其研究成果不仅可用于建造基于知识的系统，而且可直接应用于图像分析、信息检索、机器人规划和模糊控制等众多领域。

1984 年，Paul R. Cohen 和 Thomas R. Gruber 把

处理知识系统中的不确定性方法归纳成3类：工程方法、控制方法和并行确定性推理方法。数值不确定性推理方法隶属于并行确定性推理方法。1988年，Judea Pearl从语义分类的角度出发，又把数值不确定性推理方法分为外延方法和内涵方法两种。外延方法是面向产生式系统(或规则系统)的不确定性推理方法，具有计算简便的优点。内涵方法是面向陈述系统(或基于模型的系统)的不确定性推理方法，具有语义清晰的优点。参考以上分类方法，下面分别介绍数值不确定性推理的外延方法和内涵方法，以及不确定性推理的工程方法和控制方法。

(1) 数值不确定性推理的外延方法

确定性因子理论(certainty factors theory)是1975年由Edward H. Shortliffe等为建造医疗诊断专家系统MYCIN而提出的，是不确定性推理中最早且最简单的方法之一。该理论采用确定因子CF(H,E)作为规则的不确定性度量，它被定义为证据E存在的前提下，对于假设H的信任和不信任之间的差；用CF(E,e)表示证据E的不确定性值，其中e表示与E有关的所有证据；不确定性推理过程就是利用证据的不确定性值和规则的不确定性值求出结论的不确定性值的过程。确定性因子理论的主要困难是有时该方法得到的CF值和条件概率值相反，且推理链中任意两条规则的确定性因子间也难以保证概率独立。但由于其不确定性值的计算简单，易于理解，信任和不信任被清晰地分开，能表达无知，所以基于确定性因子理论的不确定性推理方法有着广泛的应用。

主观贝叶斯方法(subjective Bayesian method)，或称主观概率论，是Richard O. Duda等于1976年提出的一种不确定性推理模型。该方法在著名的地质勘探专家系统PROSPECTOR中得到了成功应用。该方法采用规则表达领域知识，每条规则有两个规则强度LS和LN，分别作为规则的充分性度量和必要性度量，用以描述规则的不确定性。若将每条规则的前提和结论表示成结点，前提和结论之间用一条有向边相连，那么多条规则即可表示为一个有向图，称为推理网络。其不确定性推理过程，就是根据证据的先验概率，利用LS或LN，将结论的先验概率更新为后验概率的过程。主观贝叶斯方法是不确定性推理的重要方法之一。主观贝叶斯方法的主要不足是难以有效处理“无知”，在无知的情况下(没有任何先验知识时)，概率论须对每一假设赋以相同的概率值。

(2) 数值不确定性推理的内涵方法

Lotfi A. Zadeh 1965年提出模糊集(fuzzy set)理论，从研究集合与元素的关系入手研究不确定性，使用隶属函数描述模糊概念，一个对象可以部分地属于一个集合，用隶属度度量其属于这个集合的程度，从而将经典二值逻辑推广到多值逻辑，使得不确定性可以用[0,1]上的区间来度量。

粗糙集理论(rough set theory)是Zdzislaw Pawlak于1982年提出的一种能够定量分析处理不精确、不一致、不完整信息与知识的数学工具，是经典集合论的推广。该理论基于等价关系，将论域中的对象分割成一些等价类，且这些等价类构成论域上的划分。论域中的子集若能够表示为这些等价类的并集，则称该子集是确定的(同时该子集也称为论域中的可定义集)，否则称该子集是粗糙的。若某个子集是粗糙的，通过其在近似空间的上、下近似算子(又称上、下近似集)来表示不确定性。其中，下近似集是包含于该子集的最大可定义集，上近似集是包含该子集的最小可定义集。粗糙集中的不确定性度量一般采用信息熵、粗糙熵、粗糙度、近似分类质量、近似分类精度和条件信息熵等作为准则。

为了有效处理由“无知”引起的不确定性，Arthur P. Dempster和Glenn Shafer提出了**证据理论**(evidence theory)。证据理论引入信任函数$Bel(A)$和似然函数$Pls(A)$分别表示对证据A的信任程度和不反对A的程度，进而用区间$[Bel(A), Pls(A)]$表示证据A的不确定性。当出现多个可用证据时，采用Dempster证据组合规则对多个证据进行组合以产生一个更好的信任评价。证据理论不要求必须对无知假设赋以信任值，因此可以有效处理“无知”。由于其证据组合规则可以综合不同来源的数据或知识，使证据理论成为信息融合的一种重要方法。证据理论的主要困难在于，不相容证据的组合。

贝叶斯网(Bayesian network)以贝叶斯方法为理论基础，自1988年由Judea Pearl提出后，就一直是人工智能领域处理不确定性问题的重要工具，并得到广泛应用。贝叶斯网是概率论与图论相结合的产物，其实质上是一种基于概率的不确定性推理网络，它是变量间联合概率分布的图形表示方式，由结构和参数两部分组成。贝叶斯网的结构是一个有向无环图，结点表示随机变量，有向边表示变量间的因果关系。贝叶斯网的参数即条件概率分布，表示变量间影响关系的强弱。基于贝叶斯网的不确定性推理就是根据给定的证据变量值，计算查询变量后验概

率的过程。贝叶斯推理方法包括精确推理算法(如变量消元算法和团树传播算法等)和近似推理算法(如随机抽样法和变分法等)。贝叶斯网早期往往由领域专家手工建造,目前的研究主要集中于通过**机器学习**方法从数据中自动获得贝叶斯网。贝叶斯网的优点在于使用图形方式表示联合概率分布,在逻辑上清晰、直观、易于理解;并且利用变量间的条件独立关系把复杂的联合概率分布分解成一系列相对简单的模块,降低了知识获取的难度和推理的复杂度。贝叶斯网面临的主要困难是其学习和推理都是 NP 难解的。

关系概率模型(relational probability models, RPMs)是 20 世纪 90 年代末出现的一种新的内涵不确定性推理模型。由于贝叶斯网的变量集是确定且有限的,并且每个变量都有确定的值域,这限制了贝叶斯网的应用。关系概率模型把概率论和一阶逻辑结合起来,能够扩展可以处理的问题范围。它通过引入实体、实体属性和实体间关系等概念对标准贝叶斯网进行扩展,比贝叶斯网具有更强的表达能力。在 RPMs 中,领域对象属于不同的类 X,每个类都有一个属性集 $A(X)$,RPMs 的图形结构由结点和有向边构成,结点之间存在的依赖关系用有向边来表示。每个结点对应类的一个属性,且有一个条件概率分布与其相关联,表示该结点之父结点对其的影响程度。指向每个结点的边有两种类型:一种为指向同一个类中与之相关的另一属性的边,另一种是指向相关类中某属性的边。RPMs 的推理及学习算法可通过扩展标准贝叶斯网的推理及学习算法得到。概率论与一阶逻辑的结合研究是不确定性推理的未来发展方向之一。

(3) 不确定性推理的工程方法和控制方法

这两种方法的特点是通过控制策略的巧妙设计,来限制或减少不确定性对系统的影响。它们没有处理不确定性的统一模型,处理效果严重依赖于控制策略。

工程方法指系统的构造避开了不确定性的影响。具体包含 3 种形式:一是预测影响系统性能的不确定性因素,然后通过将问题形式化消去这些不确定性因素;二是相关性假设,它将不确定性排除在人工智能系统之外;三是考虑领域的不完备模型所引起的不确定性。

控制方法通过识别领域中引起不确定性的某些特征,再利用控制策略减少不确定性的影响或阐明它的来源。人工智能系统用各种方式在其控制策略中使用关于不确定性的知识。通过识别不确定性的入口,控制策略能提供某种手段取消不正确的结论和对可疑问题进行仔细分析。主要的控制策略包括:相关性制导的回溯、最小冒险计划、机缘控制和启发式查找等。

不确定性推理一直是人工智能的主要研究方向,历经 40 余年的发展,已形成了多种不确定性处理方法,每种方法既具优势又有不足,各种方法之间既有区别又相互联系。针对现有方法的不足进行改进和扩展,融合现有多种不确定性推理方法以弥补单一方法的局限性,针对新的应用需求提出新的不确定性推理方法,以及进一步拓展不确定性推理的应用范围,是不确定性推理研究的主要发展趋势。

参考文献

1. 刘大有,等. 知识系统中不确定性和模糊性处理的数值方法. 长春:吉林大学出版社, 2000

2. Halpern J Y. Reasoning about uncertainty. Cambridge, MA: MIT Press, 2005

3. Kruse R, Schwecke E, Heinsohn J. Uncertainty and vagueness in knowledge based systems. Berlin: Springer-Verlag, 1991

4. Russell S, Norvig P. Artificial intelligence: A modern approach. 3rd ed. Upper Saddle River, NJ: Prentice Hall, 2009

(刘大有 杨博)

Buer daishu

布尔代数(Boolean algebra) 设集合 B 含特定元素 0 和 1,若 B 上的一元运算"$-$"和二元运算"$+$"与"$\cdot$"满足

(1) $x+y=y+x$ 且 $x\cdot y=y\cdot x$ (交换律)

(2) $(x+y)+z=x+(y+z)$ 且 $(x\cdot y)\cdot z=x\cdot(y\cdot z)$ (结合律)

(3) $x+(y\cdot z)=(x+y)\cdot(x+z)$ 且 $x\cdot(y+z)=x\cdot y+x\cdot z$ (分配律)

(4) $x+0=x$ 且 $x\cdot 1=x$ (零律)

(5) $x+\bar{x}=1$ 且 $x\cdot\bar{x}=0$

其中 $x,y,z\in B$,则称 B 为一个布尔代数。例如在集合{0,1}上定义运算 –. eps, + 和 ·,如下列表格所示:

+	0	1
0	0	1
1	1	1

·	0	1
0	0	0
1	0	1

–	
0	1
1	0

即可获得一个布尔代数，通常称为逻辑代数。此外，命题代数和集合代数也都是常用的布尔代数。

布尔代数是由英国数学家 G. Boole 在 1854 年首先提出来的，是为了把逻辑推理变为代数演算。到了 20 世纪 30 年代，随着计算机科学的飞速发展，布尔代数的近代理论才开始形成，并获得广泛的应用。M. H. Stone 在 1936 年证明了任意一个布尔代数都同构于一个集合代数。任意一个布尔代数也都同构于一个命题代数。此外还有原子布尔代数同构于该原子集合上的集合代数；完全原子布尔代数同构于该原子集合上的幂集代数。

一个布尔函数的最简式是指表示该函数的在某种意义下最简的多项式表达式，通常是指项数最少，或者文字个数最少，它与组合逻辑线路的最佳设计密切相关。根据字母数的多少可分别采用奎因方法和卡诺图法求出其最简式。对于 2 值布尔函数，E. L. Post 在 1921 年证明了一个布尔函数系能生成所有布尔函数之充要条件为该系中含有非保 0、非保 1、非自对偶、非线性和非单调 5 类函数，即著名的完备性定理。

设 $f(x_1,x_2,\cdots,x_n)$ 是布尔函数，一般 n 元布尔方程 $f(x_1,x_2,\cdots,x_n)=0$ 有解的充要条件为 $\prod_{\substack{\alpha_i=0,1\\ i=1,2,\cdots,n}} f(\alpha_1,\alpha_2,\cdots,\alpha_n)=0$。采用逐次消元法可求出其特解、通解、再生通解和无冗余通解。

布尔代数中的 n 个元素 $c_1,c_2,\cdots,c_n$ 是正规的，如果 $\sum_{i=1}^{n} c_i=1$；是直交的，如果 $c_ic_j=0, i\neq j, i,j=1,\cdots,n$；是正交，如果正规且直交。线性布尔方程 $\sum_{i=1}^{n} a_ix_i=0$ 有正交解的充要条件为 $\prod_{i=1}^{n} a_i=0$，其正交参数解（通解、再生通解）为 $x_i=\sum_{j=1}^{m} b_{ij}t_j, i=1,\cdots,n$，这里 $b_{1j},\cdots,b_{nj}$ 正交，$j=1,\cdots,m$；$\sum_{j=1}^{m} b_{ij}\leqslant \bar{a}_i, i=1,\cdots,n$。

布尔矩阵有逆的充要条件为每一行正规（直交）且每一列直交（正规），其逆为其转置矩阵。矩阵 $\boldsymbol{G}$ 称为矩阵 $\boldsymbol{A}$ 的一个广义逆，如果 $\boldsymbol{A}=\boldsymbol{AGA}$。根据不同的用途，广义逆又有下列几种：极小范数逆，$\boldsymbol{GA}=(\boldsymbol{GA})^{\mathrm{T}}$，其中 $(\boldsymbol{GA})^{\mathrm{T}}$ 是 $\boldsymbol{GA}$ 的转置矩阵；最小二乘逆，$\boldsymbol{AG}=(\boldsymbol{AG})^{\mathrm{T}}$；梯林-瓦格纳逆，$\boldsymbol{G}=\boldsymbol{GAG}$。

在布尔代数的定义中从左（右）边等式内将运算 + 与运算 · 互换、0 与 1 互换后便可得右（左）边等式，即具有对偶性，从而布尔代数中的任意公式均具有对偶性。若在布尔代数中引入代数运算 $\oplus$：$x\oplus y=x\bar{y}+\bar{x}y$，则布尔代数便成为一个环。这样，布尔代数的理论与此特殊类型环的理论等价。由于布尔代数对逻辑规律进行了数学处理，又在线路设计和自动化系统中有广泛应用，故也称为逻辑代数、开关代数等。

参考文献

1. Rudeanu S. Boolean functions and equations. Amsterdam：North-Holland，1974

2. Monk J D，Bonnet R. Handbook of Boolean algebras. Vol. 1～3. Amsterdam：North-Holland，1989

3. 金基恒. 布尔矩阵理论及其应用. 何善堉，等译. 北京：知识出版社，1987

（罗铸楷）

butu jishu

布图技术（layout technology） 按照工程要求把已设计的电路图转变为工艺图或掩膜图的设计技术。布图设计主要包括布局和布线。布局是将元件几何图形的布置优化；布线是按优化规则将电气上相通的点连通。

印制电路板的设计是将电子系统的电路图变为布线图的布图设计。印制电路板上的元件几何图形及连线图统称为导电图形。其设计层数分为单层、双层及多层。其设计规则与所用的元件电气性能及加工工艺要求有关，常用的有最小线宽，线与线、线与焊盘、焊盘与焊盘的最小间隙，最小走线长度，焊盘的最小尺寸等。

集成电路的布图设计称为版图设计。集成电路的制造工艺复杂，光刻是它的关键工艺之一。每次光刻都要用掩膜版。因而集成电路的版图设计就成了集成电路设计和制造的关键步骤。集成电路的版图设计规则，是设计人员在绘制各层几何图形时应遵守的一些限制。如最小宽度、最小间隔、最小延伸、最小重叠以及不同层的导体接触规则，如接触孔形状、尺寸、互补金属-氧化物-半导体（CMOS）与 N 沟道金属-氧化物-半导体（NMOS）规则等。

在 20 世纪 60 年代，布图设计由手工进行。人工设计周期长，效率低，易出错。随着计算机图形显示技术的进步和发展，在 70 年代后期开始应用计算机辅助设计技术，出现版图半自动和自动设计系统。

计算机辅助版图设计是根据集成电路的性能要求，先人工布局，在坐标纸上绘制出以电路几何图形

表示的总图的草图。然后用符号布局语言、版图图形编辑语言或人机交互版图设计系统，将电路几何图形编排在版图上，再把总图分层，由与计算机相连的外部设备和专用设备（如光学图形发生器、自动刻图机）完成绘制总图、刻红模、制作初编版和精缩版的工作。符号布局语言是用一些符号定义几何图形，指定层属性、图形坐标位置值、图形尺寸值、图形重复间距及重复系数等。版图图形编辑语言除能直接描述版图的几何图形（如矩形、多边形、连线、圆及圆弧），还能提供图形编辑语句，如插入、平移、镜像、旋转、删除、扩大、缩小、局部移动和重复等处理。人机交互版图设计为设计人员提供良好的设计环境。设计人员使用**计算机辅助设计**（CAD）系统提供的一些简明的选单，能直接在图形显示器上生成和修改集成电路的布局设计，包括画草图、对单元设计进行压缩、把单元图定义为符号、重复安置、确定芯片几何尺寸、对设计差错及违反设计规则情况进行检查等。

版图自动设计是指芯片上的几何图形由 EDA（参见**电子设计自动化**）系统自动完成。设计工程师只要给出片子的逻辑描述，系统会自动地按照给定的逻辑描述产生芯片版图。版图自动设计方法分为门阵列、宏单元阵列、标准单元阵列、任意单元、多芯片等，分别说明如下：

（1）门阵列设计方法　门阵列设计技术是利用预先制造好的“母片”来进行布图设计。母片中央部位以一定的间距成行成列地排列着基本单元电路。基本单元电路由 6～10 个晶体管组成，基本单元内的电阻、晶体管是预先制造好的。它的内部元件排列及其大小形状都相同。对基本单元内部元件进行不同的连线就可以构成如门、触发器、加法器等。母片的四周是输入输出单元，用作输入输出电平转换、输入保护、输出功率驱动等。芯片的最外围是焊点区。用门阵列实现一个逻辑系统需要逻辑变换、单元分配、自动布线等步骤。

（2）宏单元阵列设计方法　把基本单元按列重复排列，基本单元之间有一个布线通道。功能简单的逻辑元件由一个基本单元构成，功能较复杂的逻辑元件由若干个基本单元相连接构成。这些逻辑元件称为宏单元。宏单元设计系统需要一个宏单元库。实现宏逻辑系统的步骤与门阵列相同。

（3）标准单元阵列设计方法　把预先设计好的功能单元按列布局。功能单元可以是触发器或功能块。要求这些功能单元的版图设计成宽度相同，而高度可以不同。需要标准单元库存放这些单元的电子和逻辑特性以及几何尺寸。

（4）任意单元设计方法　这是一种定制电路设计方法，又称积木块法。它利用预先设计好的功能单元的版图（门电路、触发器、寄存器、随机存储器、只读存储器等）进行电路的布图设计。要求单元版图的形状是正交图形，但大小任意。单元引出线允许在单元的四边。在布图设计时，根据逻辑图的互连要求和单元的外形参数进行单元安置，留出布线通道区，通过合理的通道布线，实现电路互连。

（5）多芯片设计方法　把多个芯片通过相互连线安置在同一个外壳封装中，这是近年来迅速发展的一种设计技术。

参考文献

1. Rabaey J M, Chandrakasan A, Nikolic B. Digital integrated circuits: A design perspective. 北京：清华大学出版社，2007

2. 曾庆贵. 集成电路版图设计. 北京：机械工业出版社，2008

（潘雪增）

buxian xitong biaozhun

布线系统标准（cabling system standard）

国内外标准化组织机构制订的与布线系统相关标准。

为满足市场对布线系统标准化的需求，规范建筑与建筑群的语音、数据、图像及多媒体业务综合网络的建设，国内外标准化组织制定了一系列布线系统的标准，用于规范布线系统的技术术语、系统结构、系统安装和测试、线缆及相关部件的参数和测试等相关内容。制订布线系统标准的国际组织主要有国际标准化委员会 ISO/IEC、北美 TIA/EIA 和欧洲标准化委员会 CENELEC。主要的布线标准有：

国际标准

ISO/IEC 11801:2002，Information technology-Generic cabling for customer premises。由国际标准化组织 ISO 制定发布的布线系统标准，该标准定义了布线系统的相关元器件参数和测试方法。

美国 EIA/TIA 标准

（1）EIA/TIA 568，Generic Telecommunications Cabling for Customer Premises，商业建筑通信布线标准。1991 年由美国电子工业协会/电信工业协会（EIA/TIA）发布，定义布线系统的线缆和相关组成部件的物理和电气指标，该标准规定了屏蔽双绞线、非屏蔽双绞线和光纤的相应参数和指标。随着技术

的进步和布线系统的发展,EIA/TIA568 标准不断增加相应的标准规范,目前版本是 ANSI/TIA-568-C(2009 年)。

(2) EIA/TIA 569,Commercial Building Standard for Telecommunications Pathways and Spaces,商业建筑通信布线施工标准。1990 年由美国电子工业协会/电信工业协会(EIA/TIA)发布,是规范建筑物内部和建筑物之间的设计和施工的标准,用于减少布线系统对不同厂商设备和介质的依赖性,目前版本是 TIA-569-B(2004 年)。

(3) TIA 606, Administration Standard for Commercial Telecommunications Infrastructure,用于提供一整套独立于各种应用之外的统一的标识和管理方案。标识管理系统独立于语音、数据、图像传输等各种应用之外,规范布线系统的标识和管理可以使得不同应用系统在增加或变更时变得更加方便,目前版本是 TIA-606-A(2007 年)。

欧洲标准

与美国标准 EIA/TIA568 和 EIA/TIA569 标准相对应的欧洲标准是 EN 50173,Information technology-Generic Cabling Systems,通用布线系统标准和 EN 50174, Information technology—Cabling Installation,线缆安装标准。

国家标准

(1)《建筑与建筑群综合布线系统工程设计规范》,GB 50311。中国制定的布线系统标准,参考国内外相关标准,规范建筑与建筑群的语音、数据、图像及多媒体业务综合网络建设,适用于新建、扩建、改建建筑与建筑群综合布线系统工程设计,目前版本是 GB 50311—2007。

(2)《综合布线工程验收规范》,GB 50312。中国制定的布线工程验收标准,用于统一建筑与建筑群综合布线系统工程施工质量检查、随工检验和竣工验收等工作的技术要求,适用于新建、扩建、改建建筑与建筑群综合布线系统工程设计,目前版本是 GB 50312—2007。

测试认证规范

(1) TIA TSB-67,用于现场安装的 5 类双绞线的认证规范,规定了通道(channel)和基本连接(basic link)的相关测试参数。

(2) TIA TSB-95,用于测试超 5 类线,以支持 IEEE 802.3 标准的 1000 BASE-T。

(3) TIA TSB-155,用于测试 6 类线,以支持 IEEE 802.3an 标准 10GBASE-T。 (尚群)

C

canshu guji

参数估计(parameter estimation) 根据从母体抽得的样本,估计母体分布中包含的未知参数。通常一个母体的分布 $F(x,\theta)$ 依赖于一个或几个参数,它们是未知的。比如正态分布 $N(\mu,\sigma^2)$ 中的参数 μ 与 σ^2,泊松分布 $P(x,\lambda)$ 中的参数 λ。现从母体 X 中抽取子样 $X_1,X_2,\cdots,X_n$,构造一个统计量 $T=T(X_1,X_2,\cdots,X_n)$ 作为对 θ 的估计,通常记为 $\hat{\theta}=T(X_1,X_2,\cdots,X_n)$,这样得到的估计称为点估计。如果 $E\hat{\theta}=\theta$,则称这个估计是无偏的。即使是一个无偏估计,也不能保证由 $\hat{\theta}$ 准确地估计出 θ。可以证明,如果 $\hat{\theta}$ 是连续型随机变量,则 $P(\hat{\theta}=\theta)=0$。用一个量估计 θ 是很难奏效的,于是就作出两个统计量 T_1,T_2,使得 $P(\omega: T_1\leqslant\theta\leqslant T_2)=1-\alpha$,其中 α 是事先给定的正数。称 $[T_1,T_2]$ 是 θ 的置信度为 $1-\alpha$ 的区间估计,T_1 与 T_2 分别称为 θ 的置信下限和置信上限。区间估计的方法参见**假设检验**。点估计方法有下列两种:

矩估计 通过用子样矩估计母体矩的方法导出的参数估计称为矩估计。给定子样 $X_1,X_2,\cdots,X_n$,称 $\overline{X}\triangleq\frac{1}{n}\sum_{i=1}^{n}X_i$ 为子样均值,$s^2\triangleq\frac{1}{n}\sum_{i=1}^{n}(X_i-\overline{X})^2$ 为子样方差,$\frac{1}{n}\sum_{i=1}^{n}X_i^r$ 为子样的 r 阶矩。用 $\overline{X}$ 估计母体的数学期望 μ,用 s^2 估计母体的方差都是矩估计。

极大似然估计 设母体 X 的密度函数为 $f(x;\theta_1,\theta_2,\cdots,\theta_k)$,其中 $\theta_1,\theta_2,\cdots,\theta_k$ 为未知参数,$X_1,X_2,\cdots,X_n$ 是来自母体 X 的子样。于是 $(X_1,X_2,\cdots,X_n)$ 的联合分布密度为 $\prod_{i=1}^{n}f(x_i;\theta_1,\theta_2,\cdots,\theta_k)$,称

$$L(\theta_1,\theta_2,\cdots,\theta_k)=\prod_{i=1}^{n}f(X_i;\theta_1,\theta_2,\cdots,\theta_k)$$

为 $\theta_1,\theta_2,\cdots,\theta_k$ 的似然函数。若有估计量 $\hat{\theta}_1,\hat{\theta}_2,\cdots,\hat{\theta}_k$ 使得

$$L(\hat{\theta}_1,\hat{\theta}_2,\cdots,\hat{\theta}_k)=\max\{L(\theta_1,\theta_2,\cdots,\theta_k)\}\tag{1}$$

这里 max 是对 $(\theta_1,\theta_2,\cdots,\theta_k)$ 所有可能的取值而言,则称 $\hat{\theta}_1,\hat{\theta}_2,\cdots,\hat{\theta}_k$ 分别为 $\theta_1,\theta_2,\cdots,\theta_k$ 的最大似然估计。直观地说,$(X_1,X_2,\cdots,X_n)$ 落在 $(x_1,x_2,\cdots,x_n)$ 的邻域里的概率是 $\prod_{i=1}^{n}f(x_i,\theta_1,\theta_2,\cdots,\theta_k)\mathrm{d}x_i$。显然,不同的 $\theta_1,\theta_2,\cdots,\theta_k$ 将会影响这个概率,因为概率越大的事件越容易发生,因此,可将观察到的 $(X_1,X_2,\cdots,X_n)$ 取值为 $(x_1,x_2,\cdots,x_n)$,由此可认为,θ_j 应该是使 $\prod_{i=1}^{n}f(x_i,\theta_1,\theta_2,\cdots,\theta_k)$ 达到了最大值。所以可用式(1)来确定 θ_i 的估计。

为了求得极大似然估计,通常是求

$$\frac{\partial}{\partial\theta_j}\ln L(\theta_1,\theta_2,\cdots,\theta_k)=0,\quad j=1,2,\cdots,k$$

的解 $\hat{\theta}_j=\hat{\theta}_j(X_1,X_2,\cdots,X_n)$,$j=1,2,\cdots,k$。

一个估计的好坏,可从不同侧面提出一些标准,如无偏性、最小方差无偏性、有效性、充分性等。

参考文献

成平,陈希孺,等. 参数估计. 上海:上海科学技术出版社,1985

(金治明)

canshu qumian

参数曲面(parametric surface) 用参数表达式定义的**曲面**。它是**参数曲线**的表示形式在参数域上的拓广。在参数曲线中介绍的几种表示形式,如 Ferguson、Bézier、B 样条、NURBS 等经扩展后均可用于表示参数曲面。

参数曲面在汽车、飞机、船舶、家用电器、建筑物、玩具等多种产品和工程的设计、制造以及动画、影视的制作中均广泛应用。

复杂曲面可由一系列的曲面片拼合而成。

实际中常用定义在矩形域上的参数曲面片,它是由曲线边界包围的具有一定连续性的点集面片。若以两个参数的单值函数表示面片,其表示式为

$$x=x(u,w)$$
$$y=y(u,w)$$
$$z=z(u,w)$$
$$p(u,w)=[x(u,w),y(u,w),z(u,w)],$$
$$u,w\in[0,1]$$

参数曲面片的常用几何元素有以下几种：

（1）角点　把 $u,w=0$ 或 1 代入 $p(u,w)$，得到 4 个角点 $p(0,0)$，$p(1,0)$，$p(0,1)$ 和 $p(1,1)$，简记为 p_{00}，p_{01}，p_{10}，p_{11}。

（2）边界线　矩形域曲面片的 4 条边界线是 $p(u,0)$，$p(u,1)$，$p(0,w)$，$p(1,w)$，简记为 p_{u0}，p_{u1}，p_{0w}，p_{1w}。

（3）曲面片上一点　该点为 $p(u_i,w_j)$，简记为 p_{ij}。

（4）p_{ij} 点的切矢　在面片上一点 p_{ij} 处有 u 向切矢为 p_{ij}^u，w 向切矢为 p_{ij}^w。

（5）p_{ij} 点的法矢　在 p_{ij} 处的法矢记为 $\boldsymbol{n}(u_i,w_j)$，简记为 $\boldsymbol{n}_{ij}$，

$$n_{ij} = \frac{p_{ij}^u \times p_{ij}^w}{|p_{ij}^u \times p_{ij}^w|}$$

在矩形域上的常见的参数曲面形式有以下几种：

（1）双三次参数曲面片　由 2 个三次参数变量 (u,w) 定义的曲面片。曲面片是参数曲面的基本单元。双三次参数曲面片是应用得最广泛的一种曲面片，其代数形式是

$$P(u,w) = \sum_{i=0}^{3}\sum_{j=0}^{3} a_{ij}u^i w^j,\ u,w \in [0,1]$$

（2）Bézier 曲面　用 Bernstein 多项式及控制点网格定义的曲面。基于 Bézier 曲线，可以给出 Bézier 曲面的表示式。设 $p_{ij}(i=0,1,\cdots,n;j=0,1,\cdots,m)$ 为 $(n+1)\times(m+1)$ 个空间点列，则 $m \times n$ 次 Bézier 曲面定义为

$$S(u,w) = \sum_{i=0}^{m}\sum_{j=0}^{n} B_{i,m}(u)B_{j,n}(w)p_{ij},\ u,w \in [0,1]$$

式中，$B_{i,m}(u)=C_m^i u^i(1-u)^{m-i}$，$B_{j,n}(w)=C_n^j w^j(1-w)^{n-j}$ 是 Bernstein 基函数。依次用线段连接点列 $p_{ij}(i=0,1,\cdots,n;j=0,1,\cdots,m)$ 中相邻两点所形成的空间网格，称为控制点网格。

（3）B 样条曲面　用分段 B 样条多项式函数及控制点网格定义的曲面。基于 B 样条曲线，可以得到 B 样条曲面的表示式。给定 $(n+1)\times(m+1)$ 个空间点列 $p_{ij}(i=0,1,\cdots,n;j=0,1,\cdots,m)$，则 $S(u,w) = \sum_{i=0}^{m}\sum_{j=0}^{n} p_{ij}N_{i,k}(u)N_{j,l}(w)$，$u,w\in[0,1]$ 定义了 $k\times l$ 次 B 样条曲面。式中 $N_{i,k}(u)$ 和 $N_{j,l}(w)$ 分别是 k 次和 l 次的 B 样条基函数，由 p_{ij} 组成的空间网格成为控制点网格。

（4）非均匀有理 B 样条曲面　用双参数非均匀有理 B 样条函数、控制点列及其与控制点列对应的权因子定义的面，也称 NURBS 曲面。

由双参数变量分段有理 B 样条多项式定义的 NURBS 曲面是

$$S(u,v) = \sum_{i=0}^{m}\sum_{j=0}^{n} W_{ij}P_{ij}N_{i,p}(u)N_{j,q}(v) \Big/ \sum_{i=0}^{m}\sum_{j=0}^{n} W_{ij}N_{i,p}(u)N_{j,q}(v),\ u,v \in [0,1]$$

式中，P_{ij} 是矩形域上控制点网格点列，W_{ij} 是相应控制点的权因子，$N_{i,p}(u)$ 和 $N_{j,q}(v)$ 分别是 p 次和 q 次的 B 样条基函数，它们都是在节点矢量 $\boldsymbol{S}[s_0,s_1,\cdots,s_{m+p+1}]$ 和 $\boldsymbol{T}[t_0,t_1,\cdots,t_{n+q+1}]$ 上定义的。如果在非均匀参数轴上定义的节点矢量 $\boldsymbol{S}$，$\boldsymbol{T}$ 具有下述形式：

$$\boldsymbol{S} = [\underbrace{0,0,\cdots,0}_{(p+1)个},s_{p+1},\cdots,s_{m+p+1},\underbrace{1,1,\cdots,1}_{(p+1)个}]$$

$$\boldsymbol{T} = [\underbrace{0,0,\cdots,0}_{(q+1)个},t_{q+1},\cdots,t_{n+q+1},\underbrace{1,1,\cdots,1}_{(q+1)个}]$$

则由 $\boldsymbol{S}$，$\boldsymbol{T}$ 定义的曲面是非均匀有理 B 样条曲面，简称 NURBS 曲面。

参考文献

1. Farin G. Curves and Surfaces for Computer Aided Geometric Design. Academic Press Inc., 1993

2. Mortenson M E. Geometric Modeling. John Wiley & Sons, 1985

3. 孙家广，等. 计算机图形学. 3 版. 北京：清华大学出版社，1998

（孙家广　冯结青）

canshu quxian

参数曲线（parametric curve）　用参数表达式定义的**曲线**。如果参数用 t 表示，则平面曲线上每一点笛卡儿坐标的参数式是：

$$x = x(t)$$
$$y = y(t)$$

该点坐标的矢量表示是：

$$\boldsymbol{P}(t) = [x(t),y(t)]$$

参数曲线的切矢量或导函数是：

$$\boldsymbol{P}'(t) = [x'(t),y'(t)]$$

在实际应用中，往往只对曲线的某一部分感兴趣。通常将参数变量规格化，使 t 在 $[0,1]$ 闭区间内变化，写成 $t\in[0,1]$。此时，称之为参数曲线段。

常见的参数曲线形式有以下几种：

（1）Ferguson 曲线　用三次参数多项式定义的自由曲线。一条三次参数曲线的矢量形式是

$$\boldsymbol{P}(t) = a_3t^3 + a_2t^2 + a_1t + a_0,\quad t \in [0,1] \quad (1)$$

由式（1），可以得到参数曲线的几何形式：

$$\boldsymbol{P}(t) = F_1\boldsymbol{P}_0 + F_2\boldsymbol{P}_1 + F_3\boldsymbol{P}_0' + F_4\boldsymbol{P}_1' \quad (2)$$

其中，P_0，P_1，P'_0，P'_1为系数，分别是曲线的两个端点 $P(0)$，$P(1)$以及对应的切矢量 $P'(0)$，$P'(1)$；$F=[F_1,F_2,F_3,F_4]$为调和函数

$$F_1 = 2t^3 - 3t^2 + 1,\quad F_2 = -2t^3 + 3t^2$$

$$F_3 = t^3 - 2t^2 + t,\quad F_4 = t^3 - t^2 \quad t\in[0,1]$$

这种由端点及其切矢量定义的三次参数曲线称为 Ferguson 曲线或 Hermit 曲线。

（2）Bézier 曲线　用 Bernstein 多项式函数和控制点构造的参数曲线。Bézier 方法将函数逼近同几何表示结合起来。

给定 $n+1$ 个空间点列 $P_i(i=0,1,\cdots,n)$，则 n 次 Bézier 曲线定义为

$$C(t) = \sum_{i=0}^{n} P_i B_{i,n}(t),\quad 0\leqslant t\leqslant 1$$

式中 $B_{i,n}(t)=C_n^i t^i(1-t)^{m-i}$，是 Bernstein 基函数，也是曲线上各点坐标的调和函数。依次用线段连接点列 $P_i(i=0,1,\cdots,n)$中相邻两点所形成的折线集，称为 Bézier 特征多边形，与每一条曲线对应。曲线的起始点和终点与该多边形的起始点、终点相重合，且多边形的第一条边和最后一条边表示曲线在起始点和终点处的切矢量方向。曲线的形状趋向于特征多边形的形状。

（3）B 样条曲线　用 B 样条函数构造的曲线。它除了保持 Bézier 曲线的直观性和凸包性等优点外，还可以进行局部修改，且曲线更逼近特征多边形。曲线的阶次也与顶点数无关，因而更方便灵活。

已知 $n+1$ 个控制点 $P_i(i=0,1,\cdots,n)$为特征多边形的顶点，k 阶$(k-1)$次 B 样条曲线的表达式为

$$C(u) = \sum_{i=0}^{n} P_i N_{i,k}(u)$$

其中，$N_{i,k}(u)$是 B 样条调和函数，也称为 B 样条基函数，按照递归公式可以定义为

$$N_{i,1}(u) = \begin{cases} 1 & 若\ t_i\leqslant u\leqslant t_{i+1} \\ 0 & 其他 \end{cases}$$

$$N_{i,k}(u) = \frac{(u-t_i)N_{i,k-1}(u)}{t_{i+k-1}-t_i} + \frac{(t_{i+k}-u)N_{i+1,k-1}(u)}{t_{i+k}-t_{i+1}},\quad t_{k-1}\leqslant u\leqslant t_{n+1}$$

其中，t_i 是结点值，结点是非减序列。当结点沿参数轴作均匀等距分布时，则为均匀 B 样条函数；反之，则表示非均匀 B 样条函数。

（4）非均匀有理 B 样条（NURBS）曲线　用非均匀有理 B 样条函数构造的曲线，也称 NURBS 曲线。

用分段有理 B 样条基函数定义的 NURBS 曲线是

$$C(u) = \sum_{i=0}^{n} W_i P_i N_{i,k}(u) \Big/ \sum_{i=0}^{n} W_i N_{i,k}(u) = \sum_{i=0}^{n} P_i R_{i,k}(u)$$

其中，P_i 是特征多边形顶点位置矢量，$N_{i,k}(u)$是 k 阶 B 样条基函数，W_i 是相应控制点 P_i 的权因子。结点矢量是

$$\boldsymbol{T}=[\underbrace{\alpha,\alpha,\cdots,\alpha}_{(k+1)个},t_{k+1},\cdots,t_n,\underbrace{\beta,\beta,\cdots,\beta}_{(k+1)个}]$$

用参数方程来表示曲线比用显式、隐式方程有更多的优越性：①有更大的自由度来控制曲线的形状；②可对曲线的参数方程直接进行几何变换（如平移、比例、旋转等），从而节省计算量；③便于处理斜率为无穷大的情形，不会因此而中断计算；④参数方程中，代数、几何相关和无关的变量是完全分离的，而且变量个数不限，这种变量分离的特点可以用数学公式去处理几何分量；⑤采用规格化的参数变量 $t\in[0,1]$，可使其相应的几何分量是有界的，而不必用另外的参数去定义其边界；⑥易于用矢量和矩阵表示几何分量，简化了计算。

参考文献

1. Farin G. Curves and Surfaces for Computer Aided Geometric Design. Academic Press Inc.，1993

2. Mortenson M E. Geometric Modeling. John Wiley & Sons，1985

3. 孙家广，等. 计算机图形学. 3 版. 北京：清华大学出版社，1998

（孙家广　冯结青）

canshu suanfa

参数算法（parameterized algorithm）　一种允许时间复杂性因特定问题参量数值增长而指数增长的**算法**。一般受实际限制，特定参数的数值较小，因而参数算法能够处理输入长度较大的实际数据。一般用于 NP-难解问题求解。参数计算是近 10 年发展起来的问题求解新方法。基于对实际工程应用中影响问题求解复杂性的若干参数研究，人们发现：对于许多 NP-难解问题的实际实例，问题的求解复杂性与问题实例中的某个或多个关键参数有关。另外，在实际应用环境下，即使给定问题实例规模很大，关键参数仍然很小，可充分有效利用“小参数”这一特性，设计时间复杂性为 $f(k)n^{O(1)}$ 的求解算法，其中 n 是问题实例的输入长度，k 是问题实例中

的关键参数,f 是 k 的函数。鉴于各实际领域诸多难解问题存在参数很小的现象,具有 $f(k)n^{O(1)}$ 时间复杂性的算法就变得实际可用。例如,个体单体型组装是生物信息学中的重要问题,人们基于问题输入长度(m 个 DNA 片段数据,m 一般 100 以上)设计了时间复杂性 $O(2^m)$ 的算法,该算法实际可用性差。该问题实例的两个参数实际数值并不大:一个片段覆盖的最大 SNP 位点数 k_1(通常小于 10);覆盖一个 SNP 位点的最大片段数 k_2(通常小于 19)。基于上述两个参数,对个体单体型相关问题建立参数化问题模型,可设计出时间复杂性为 $O(nk_2 2^{k_1} + m\log m + mk_1)$ 的参数算法,这样的算法由于"小参数"特性而变得实际可用了。

基本内容

从本质上讲,参数计算理论对问题的计算复杂性重新进行了划分,即将 NP-难解问题划分为固定参数可解(问题可在 $f(k)n^c$ 时间内被求解,其中 n 为问题实例的长度,k 为参数)和固定参数不可解两种类型,其框架主要包括核心化、参数算法设计与分析技术和固定参数不可解理论等,具体内容如下。

(1) 核心化(kernelization)

核心化是参数算法设计的一个主要研究方向,基本思想是通过提出一系列的简化规则,有效降低问题的规模,使得简化后问题实例的大小仅与问题实例的参数 k 有关,并且简化前后的实例保持解的一致性。

常见的核心化技术有 NT 定理、皇冠分解、极值归纳、随机方法等。基于这些技术得到了许多 NP-难解问题的核,为诸多 NP-难解问题的求解提供了有效的算法预处理方法,如点覆盖问题大小为 $2k$ 的核,反馈顶点集问题大小为 $O(k^2)$ 的核,等等。

(2) 参数算法设计与分析技术

自参数计算提出以来,人们提出了彩色编码(color-coding)、局部贪婪(greedy localization)、图分离(graph separation)、树分解(tree decomposition)、迭代压缩(iterative compression)等参数算法设计技术,并得到了许多 NP-难解问题的有效参数算法,如点覆盖问题时间复杂性为 $O(1.274^k + kn)$ 的参数算法,反馈顶点集问题时间复杂性为 $O^*(3.83^k)$ 的参数算法,k-path 问题时间复杂性为 $O^*(4^k)$ 的参数算法等。利用平摊分析(amortized analysis)和度量治之(measure & conquer)技术,能够对参数算法时间复杂性给出更精确的分析。

(3) 固定参数计算复杂性

在参数计算领域,固定参数计算复杂性理论进一步说明了为什么许多问题不可能被有效求解,即不存在 $f(k)n^c$ 时间的参数算法(n 为问题实例长度,k 为参数)。固定参数不可解理论的框架具有一定的层次结构性,通常用 W 框架表示,即 $\text{FPT} \subseteq W[1] \subseteq W[2] \subseteq \cdots W[t] \subseteq \cdots$,不同层次结构类中的问题具有的难度也不一样。

(4) 参数计算的应用

参数算法为解决实际工程应用中出现的计算难题求解提供了一种新途径,例如,在苏黎世瑞士联邦工学院计算生物化学研究组的达尔文项目中,基因序列冲突问题被转化成点覆盖问题(vertex cover),即求给定图中是否存在一个规模为 k 的点覆盖。以其生物应用所需,点覆盖的规模 k 不超过 60。由此应用出发,人们对点覆盖算法进行了深入的研究,提出了时间复杂性为 $O(1.274^k + kn)$ 的参数算法,其中 n 是给定图的大小,有效求解了生物计算中的基因序列冲突问题。另一个例子是程序类型检测问题(ML type-checking),这一问题是指数时间难解的。利用参数计算理论,对于长度为 n、嵌套深度为 k 的程序,程序类型检测问题可在 $O(2^k + n)$ 时间内被求解。注意,在一般计算科学中,程序的嵌套深度一般不超过 10,利用这一参数算法,该问题在实际中得以解决。

参考文献

1. Downey R G, Fellows M R. Parameterized complexity. Monographs in Computer Science. New York: Springer, 1999

2. Flum J, Grohe M. Parameterized complexity theory. New York: Springer, 2006

3. Niedermeier R. Invitation to fixed-parameter algorithms, volume 31 of Oxford Lecture Series in Mathematics and Its Applications. Oxford: Oxford University Press, 2006

4. Chen J. Parameterized computation and complexity: a new approach dealing with NP-hardness. Journal of Computer Science and Technology, 2005, 20: 18-37

(王建新)

caozuo xitong

操作系统(operating system) 管理系统资源,控制程序执行,改善人机交互,为其他软件系统提供支持的一种系统软件。操作系统是**软件系统**的核

心，是最靠近硬件的一层软件，它把硬件裸机拓展成为一台更加完善的虚拟机器，使得计算机系统的资源利用率更高，运行环境更好，使用和管理更加方便。

形成与发展

操作系统经历了从无到有、从简单到复杂的发展历程。第一代计算机速度慢、外设少、规模小，程序员可以直接通过手工操作控制程序的执行。随着第二代计算机的出现，手工操作既难以进行复杂的控制，还会严重降低计算机的使用效率，于是，设计一种系统软件来管理硬件资源和控制程序执行的需求愈显迫切。操作系统就是在这种需求下，于20世纪60年代初期产生的。初期的操作系统称为监督程序或管理程序，其主要功能是提供设备驱动和输入输出控制、接收并执行操作员输入的命令和向操作员显示各种信息等。20世纪60年代，中断和输入输出通道技术的出现和发展，为实现中央处理器和外部设备的并行工作提供了支撑，操作系统开始具有多道程序和分时操作的功能。至此，操作系统开始形成。其中，最有影响力并称为现代操作系统先驱的是1964年由美国贝尔实验室和麻省理工学院等研发的大型机分时操作系统MULTICS。

进入70年代，操作系统的功能渐趋完善，针对各类规模体系的操作系统开始涌现，除MULTICS外，大型机上还有MVS，中小型机上有VM370、DOS/VSE等。80年代开始，伴随微型计算机问世，出现了XENIX，MS-DOS等微机操作系统。与MULTICS有很深渊源的UNIX是70～80年代流行最广的操作系统，对整个软件技术和产业发展都产生了深远影响。它是由美国贝尔实验室的Ken Thompson和Demis Ritchie于1969年研制的，具有功能多样，使用方便、支持交互等特点。与此同时，他们还设计并实现了一种高级程序设计语言C，UNIX的绝大部分程序用C语言书写。这就使得UNIX易于移植，可被各种型号计算机采用。90年代以来，微软公司推出的视窗操作系统Windows3.1，Windows NT，Windows XP，Windows 7等各种系列的**Windows操作系统**，不仅是近20多年来桌面计算机使用最广的操作系统，而且彻底结束了系统软件依附于硬件而存在的历史。从此，软件成了信息化的核心和独立发展的朝阳产业。90年代初，Linus Torvalds开发的Linux成为操作系统发展史上又一个里程碑，促成了操作系统的开源化发展，迄今，Linux已拥有庞大的用户群和广泛的应用领域，成为目前与UNIX和Windows并驾齐驱的主流操作系统之一。

从70年代中期到90年代，还开发了多种面向网络环境的操作系统。70年代中，为适应计算机网络发展，出现了具有统一控制的网络操作系统。80年代，为满足分布式系统和大规模并行系统的需要，出现了分布式和并行操作系统。90年代以来，随着智能化、普适化技术的快速发展，嵌入式操作系统得到迅猛发展。当前，随着信息网络技术和云计算等软件服务工程的发展，云操作系统正成为人们关注和研究的热点。

分　类

操作系统分类归纳如下：①**批处理**操作系统，支持多个用户程序同时运行于一台计算机上，在批处理操作系统支持下，计算机可同时接受多个用户程序并成批地处理。②分时操作系统，支持多个联机用户在各自的终端同时使用一台计算机，每个用户感到好像自己使用一台独立的计算机，在分时操作系统支持下，计算机为多个用户共享。③实时操作系统，为诸如工业过程控制等实时系统配置的操作系统，强调实时特性，即必须满足系统对时间的限制和要求。④网络操作系统，为计算机网络配置的操作系统，支持各台计算机通过网络进行相互通信、共享资源、文件传输和远程作业录入。⑤分布式操作系统，在分布式系统范围内实现对多台计算机的资源管理、任务分配、并行地运行分布式程序。⑥并行操作系统，为大规模并行处理系统配置的操作系统。⑦**嵌入式操作系统**，指嵌入在家用电器、手持设备和各种机电装置等系统中的操作系统。⑧多核操作系统，随着多核处理器的发展，多核计算机逐步走向应用，已经研制出多款多核操作系统。

以上分类并非绝对，且互有交叉，也还有尚未列入的，如**安全操作系统**、**多媒体操作系统**等，还有尚未实用化的，如紧耦合的网络操作系统和分布式操作系统。目前，操作系统绝大多数是在网络环境下运行的，因而又有服务器操作系统和桌面操作系统之分。

基本内容

操作系统在计算机系统中的作用有：对系统资源进行管理，对程序执行提供控制，对用户提供友好的人机交互以及对其他软件提供丰富的功能支持。

计算机系统资源包括硬件资源和软件资源。硬件资源指组成计算机的硬设备，如中央处理器、主存储器、磁带存储器、打印机、显示器、键盘输入设备等。软件资源主要指存于计算机中的各种数据和程

序。系统的硬件和软件资源都由操作系统根据用户需求和系统状态进行分配和调度。其功能主要包括:①处理器管理　根据一定调度策略将处理器交替地分配给系统内等待运行的程序。处于等待态的程序只有在获得处理器后才能运行;程序运行中若遇到某个事件(如启动外部设备)而暂时不能继续运行,处理器管理就要处理相应事件,并将处理器重新分配。常用的处理器分配和调度策略有优先数法和时间片法。采用优先数法时,等待运行的程序都指派一个优先数,处理器管理按“先大后小”的原则,将处理器分配给等待运行中的优先数最大的程序。采用时间片法时则是分配给每道程序一个时间片,若一道程序的连续运行时间超过指派的时间片时,就将其强行替换下来,排入等待运行程序队列的队尾,让队列的第一个程序占用处理器。②存储管理　负责把主存单元分配给待执行程序,执行结束后将其占用的主存单元收回以便再使用。对于提供虚拟存储器的计算机系统,操作系统采用页面调度机制,根据运行程序的要求和系统的负载情况分配主存来存放页面。③设备管理　负责分配和回收外部设备,控制外部设备按用户程序要求执行相关操作。对非存储型外部设备(如打印机),可以直接分配给一个用户程序,在使用完毕后回收以便给另一个需要的用户使用。对于存储型外部设备(如磁盘、磁带),则提供存储空间给用户存放各种程序和数据。④文件管理　向用户提供创建、撤销、读写、打开和关闭文件等功能。通过文件管理,用户可按文件名存取数据而无须知道数据存放位置,这样既便于用户使用又有利于数据共享。⑤网络与通信管理　实现网上资源共享,管理用户对资源的访问,保证信息资源的安全性和完整性;通过通信软件,按照通信协议的规定,完成网络上计算机之间的信息传送等。

操作系统的程序控制功能主要是控制用户程序的执行。操作系统控制用户程序的执行主要有以下内容:调入相应的编译、汇编和链接程序,将用某种程序设计语言编写的源程序翻译成计算机可执行的目标程序,分配主存等资源将程序调入主存并启动,按用户指定要求处理执行中出现的各种事件,协调有关意外事件的处理等。

操作系统人机交互部分的主要作用是控制有关设备的运行,理解并执行通过人机交互设备传来的各种命令和要求,人机交互功能主要靠人机交互设备和人机交互软件环境来完成。人机交互硬件设备主要有:键盘显示器、鼠标、各种模式识别设备等。早期的人机交互设施是键盘显示器,随着计算技术的发展,出现了语音输入设备、文字读入设备和图形、图像扫描输入设备,人机交互功能越来越强。人机交互软件环境指人机的软件桌面模式,早期操作系统通常基于命令行(文本模式)形式为用户提供输入输出窗体,现代操作系统通常提供良好的图形化操作界面支持用户与系统交互。

操作系统对其他软件的支持功能主要通过系统调用和应用编程接口(API)实现。为方便程序员编写或编辑其他软件,操作系统以应用编程接口的形式提供功能丰富的公共服务程序,使得各类用户可编出功能强大的其他软件。这样,从程序员角度,所使用的计算机不再是原始的硬件裸机,而是经过操作系统改造后的功能更加强大的一台虚拟器。

结构和趋势

操作系统一般都采用内核加程序模块的结构。内核是操作系统的核心部分,它常驻主存,基本功能有中断处理、处理器调度和原语管理。为了防止内核受到应用程序侵害,根据执行程序对资源和机器指令的使用权限,操作系统将处理器设置成不同状态:核心态和用户态。处理器处于核心态时,程序具有访问硬件设备和全部主存空间的权限。处理器处于用户态时,被限于当前处理器上执行程序的地址空间中,这样就使操作系统得到了保护。根据内核的实现形式,操作系统的内核结构可分为单内核和微内核。单内核是一个大二进制映象,操作系统模块间的交互通过直接调用其他系统模块中的内核函数来实现。用户则通过系统调用让处理器从用户态切换到核心态,执行相应内核函数,为应用程序提供服务。UNIX 和 Linux 都是单内核操作系统,但 Linux 引进加载和卸载模块机制,可动态装入和删除文件系统或设备驱动程序,解决了单内核功能的适应性和可伸缩性问题。**微内核**结构将原来的内核分成内核和服务器两部分。微内核结构中只有内核运行在核心态,仅提供资源控制和通信机制等系统基本功能。支持文件管理,存储管理等功能的服务器以及客户程序运行在用户态。当某客户程序请求服务器工作时,将请求参数以消息形式经内核通信机制转交服务器处理,后者完成所请求的工作并通过发送消息返回结果。微内核结构提高了系统的可靠性、灵活性、可扩充性,且适用于分布式计算模式。

目前,操作系统正在从单机环境不断地向集群环境乃至开放的网络和互联网环境拓展,其网络化

的发展趋势越来越显突出，云操作系统已成为人们关注的热点，伴之而来的安全性问题也备受人们关注。除此之外，操作系统的发展趋势还有：①开源化 采用开源软件模式不但会改变将来操作系统的开发模式，而且更重要的是会改变操作系统的使用方式。②轻量化 通用操作系统过于庞大和复杂。随着计算设备的微型化与泛在化，面向智能终端设备的轻量级操作系统将受到广泛关注。③虚拟化 随着构件化、模块化技术的不断成熟和发展，操作系统呈现出多平台统一的发展趋势，即同一个操作系统通过构件技术可灵活地进行扩展和变化，既支持桌面系统，也支持嵌入式系统，甚至支持数据中心。④人性化 随着多媒体技术及人机交互技术的发展，操作系统将提供更为绚丽的桌面和多维视觉效果，使人机界面更为人性化。

参考文献

1. Silberschatz A. Operating system concepts. 6th ed. John Wiley & Sons Inc, 2002

2. Brown R L, Denning P J, Tichy W F. Advanced operating system. IEEE Computer, 1984, 17(10)

3. 孙钟秀，等. 操作系统教程. 4 版. 北京：高等教育出版社，2008 （费翔林 叶保留 骆斌）

caozuo yuyi

操作语义（operational semantics） 用解释执行程序的抽象机器定义语言的**语义**。

计算机语言的实现是在具体的计算机系统中按照语言的语义编制编译程序，将语言中各个成分翻译成计算机系统中相应的一组操作。语言在计算机系统中的一种实现一旦完成，则对这个计算机系统而言，语言各个成分的含义也就完全确定了。因此语言的实现也可用来定义语言的语义。

操作语义的基本思想来源于**程序设计语言**的实现。1964 年 1 月英国 P. J. Landin 使用“栈—环境—控制—存储”抽象机器（简称 SECD 机器），第一次系统、严格地陈述了表达式的操作语义。

IBM 公司的维也纳实验室在 20 世纪 60 年代研究程序设计语言 PL/1 的形式定义时，提出描述操作语义的一种元语言——维也纳定义语言（简称 VDL）。1974 年欧洲计算机制造商联合会和美国国家标准局正式建议使用 VDL 定义的 PL/1 语义作为 PL/1 的标准。

1980 年前后，英国爱丁堡大学的计算机科学家提出结构式操作语义。它在一般的数学结构（不必是抽象机器）上用数学的归纳关系建立语义的解释系统。这种方法具有指称语义的结构式特征，并更多地略去了机器操作的细节。

程序设计语言中的算术表达式用于加工计算机系统中现存的数据，以形成新的数据。如表达式 $[(x_1 \times x_2)+1]$ 表示将计算机系统中对应变量 x_1，x_2 的存储单元数据相乘，再将乘积加 1，形成一个新的数据。计算机系统中保存数据的存储区可以用数据向量来表示。k 个存储单元保存的数据都是 1 时，可用 k 维向量（1，1，…，1）表示。计算机系统中还要有保存程序的区域，以及保存加工过程中必要信息的工作区。就此可提出一种解释执行算术表达式求值的抽象机器的模型，这个机器的存储区分成三部分：栈区 st（用作工作区），环境区 s（保存数据向量等），控制区 c（保存程序）。整个存储区记作 (st, s, c)，称为抽象机器的一个大状态。这个抽象机器具有识别符号、完成算术和逻辑运算、转储信息、实现大状态之间的转移等基本功能。

这个机器的大状态转移规则分为四类：

（1）$(st, s, (e_1 \text{ op } e_2)/c) \Rightarrow (st, s, e_1/e_2/\text{op}/c)$

（2）$(st, s, \boldsymbol{n}/c) \Rightarrow (\boldsymbol{n}/st, s, c)$

（3）$(st, s, x_i/c) \Rightarrow (x_i/st, s, c)$

（4）$(\boldsymbol{n}/\boldsymbol{m}/st, s, \text{op}/c) \Rightarrow (k/st, s, c)$ $(k = \boldsymbol{m} \text{ op } \boldsymbol{n})$

第一类规则表示，当控制区中待执行的程序要求完成表达式 $(e_1 \text{ op } e_2)$ 的求值时，抽象机就转移自己的大状态，准备先求子表达式 e_1 和 e_2 的值，然后再按照相应的运算 op（+，-，×或其他算子），求出整个表达式的值，符号“/”用于分割存放的信息。第二类规则表示，当求值的表达式是一个常量时，则其值就是抽象机中表示这个常量的相应的量（粗体用来区别语言中的符号和在抽象机中的相应表示），表达式的值暂存于栈区。第三类规则表示，当表达式是一个变量时，其值就是环境区中相应单元的当前值，即第 i 个变量 x_i 的值就是数据向量 $\boldsymbol{s}$ 的第 i 个分量 s_i 的值。第四类规则表示，当运算 op 的两个操作数已经求得，则可按照抽象机中的相应运算求出 op 作用于操作数的结果。

在这个抽象机中，表达式 $(x_1 \times x_2)+1$（在 x_1，x_2 值为 2 和 3 时）的求值是由下述大状态的转移序列完成的，转移符号⇒的上方标有实现这一转移依据的转移规则号，设 $s=(2,3)$。

$(st, s((x_1 \times x_2) + 1) / c)$

$\overset{(1)}{\Rightarrow}(st, s(x_1 \times x_2) / 1 / + / c)$

$\overset{(1)}{\Rightarrow}(st, s, (x_1 / x_2) \times 1 / + / c)$

$\overset{(3)}{\Rightarrow}(2 / st, s, x_2 / \times 1 / + / c)$

$\overset{(3)}{\Rightarrow}(3 / 2 / st, s, \times / 1 / + / c)$

$\overset{(4)}{\Rightarrow}(6 / st, s, 1 / + / c)$

$\overset{(2)}{\Rightarrow}(1 / 6 / st, s. eps, +, / c)$

$\overset{(4)}{\Rightarrow}(7 / st, s, c)$

这个抽象机正确刻画出算术表达式求值的全过程,故可作为算术表达式的操作语义。

为定义赋值语句($x_i := e$)的操作语义,可在上述抽象机中添加如下转移规则:

(5) $(st, s, (x_i := e); / c) \Rightarrow (st, s, e / x_i := / c)$

(6) $(\boldsymbol{n} / st, s, x_i := / c) \Rightarrow (st, s|_i^n, c)$

第五类规则表示,当抽象机执行($x_i := e$)时,先求出表达式 e 的值,然后再给 x_i 赋值。

第六类规则表示,对 x_i 赋以值 $\boldsymbol{n}$,相当于将数据向量 s 的第 i 个分量(即保存 x_i 当前值的存储单元)改为 $\boldsymbol{n}$,表示为 $s|_i^n$。

在定义程序设计语言的过程语句的操作语义时,由于不同的过程使用不同的环境,不同的栈区,不同的控制区,故引入外存储区,外存储区中保存有当前未调用的所有过程的全部环境区、栈区和控制区,这种机器就是 Landin 的 SECD 抽象机。

人们希望语言的语义是独立于实现的,故操作语义中使用了抽象机器,但仍不可避免地涉及语言的实现,这是它的弱点。但同时又是它的优点,因为设计一个系统的过程可看作是一个求精的过程,将不可执行的功能描述,逐步求精为可执行的程序,而操作语义方法在求精过程的后期,可发挥其特有作用,可用于描述更多的执行细节。

参考文献

周巢尘. 形式语义学引论. 长沙:湖南科技出版社,1985

(周巢尘　李晓山)

cengci shujuku

层次数据库(hierarchical database)　采用层次模型的**数据库**。

层次模型用树形结构表示各类实体及其间的联系。在树形结构中:

(1) 有且仅有一个节点没有父节点,这个节点称为根节点;

(2) 其他节点均有且仅有一个父节点。

在层次模型中,每个节点表示一个记录类型(简称记录型),节点之间的连线表示记录型间的联系,这种联系只能是一对多联系(包含一对一联系)。每个记录型可包含若干个字段。记录型用来描述实体,字段用来描述实体的属性。

在层次数据库中,每个记录只有一个父记录(根记录除外),即从一个记录到其父记录的映射是唯一的,所以对于每一个记录,只需指出它的父记录就可以表示出层次模型的整体结构。层次数据库中,任何一个给定的记录值只有按其路径查看时,才能显示出它的全部意义,没有一个子记录值能够脱离父记录值而独立存在。

在层次模型中,具有同一父节点的子节点之间互称兄弟节点,没有子节点的节点称为叶节点。

图 1 是一个层次模型的例子,其中学校为根节点;系和研究所是学校的子节点,它们互为兄弟节点;教研室和班级是系的子节点;教研室、班级和研究所是叶节点。

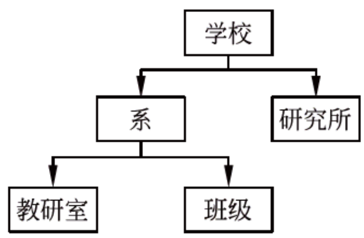


图 1　层次模型实例

储存层次数据库不仅要储存数据本身,还要反映出数据之间的层次联系,实现方法有两种:

(1) ***邻接法***　按照树的某种遍历的顺序(例如深度优先、广度优先)把所有记录值依次邻接存放,即通过物理空间的位置相邻来实现层次顺序。

(2) ***链接法***　用指针来实现数据之间的层次联系。每个记录设两类指针,分别指向最左边的一个子节点(每个记录型对应一个)和最近的兄弟节点。这种链接方法称为子—兄弟链接法。另一种链接方法是按树的先根(次序)遍历的顺序链接各记录值,称为层次序列链接法。

信息管理系统 IMS 是最早的层次数据库管理系统,由 IBM 公司研制,先后推出了多个版本。1968 年的 IMS-1 支持 HSAM 和 HISAM 存储结构;1971 年的 IMS-2 在 IMS-1 的基础上又增加了

HDAM、HIDAM 存储结构和逻辑数据库；1974 年的 IMS/VS 又增加了批处理检查点、重新启动、并发操作、辅助索引等功能。

现实世界中许多实体之间的联系本来就呈现出一种自然的层次关系，如行政机构、家族关系等。用层次模型对具有一对多层次关系的部门进行描述非常自然、直观、容易理解。这是层次数据库的突出优点。

层次数据库也存在一些弱点。在层次数据库中，父记录与子记录之间只能反映一对多的联系，而现实世界中很多联系是非层次性的，如多对多联系、一个节点可以具有多个父节点等。层次模型表示这类联系的方法很笨拙，只能通过引入冗余数据(易产生不一致性)或创建非自然的数据组织(引入虚拟节点)来解决。另外，层次数据库用存取路径来表示数据之间的联系，用户对数据的存取必须按照明确定义了的存取路径进行，必须清楚地了解数据在数据库中的当前位置，加重了用户的负担；对数据的操作是一次一个记录的存取方式，程序和数据虽有较高的物理独立性，但逻辑独立性不高。

层次数据库系统在 20 世纪 70 年代与 80 年代初非常流行，在数据库系统产品中占主导地位，虽然近年来逐渐被**关系数据库**系统取代，但在美国、加拿大等国家，由于历史原因，目前层次数据库的使用仍很多。

参考文献

1. 王珊，萨师煊. 数据库系统概论. 4 版. 北京：高等教育出版社，2006

2. Date C J. An introduction to database system. 8th ed. Reading: Addison Wesly Publishing Company, 2005

(王珊　周龙骧　王翰虎)

chazhi

插值(interpolation)　关于如何寻找函数 φ 来近似地代替任意函数 f，并使得 φ 和 f 在若干点上具有相同值的方法与过程。

插值的作法是：在事先选定的一个由简单函数所构成的含 $n+1$ 个参数 $c_0, c_1, \cdots, c_n$ 的函数类 $\varphi(c_0, c_1, \cdots, c_n)$ 中求出满足条件

$$p(x_i) = f(x_i) \quad i = 0,1,\cdots,n \qquad (1)$$

的函数 $p(x)$，并以 $p(\bar{x})$ 作为 $f(\bar{x})$ 的估值。此处，函数 $f(x)$ 称为被插值函数；$x_0, x_1, \cdots, x_n$ 称为插值结点；$\varphi(c_0, c_1, \cdots, c_n)$ 称为插值函数类；式 (1) 称为插值条件。$\varphi(c_0, c_1, \cdots, c_n)$ 中满足插值条件 (1) 的函数 p 称为插值函数，误差 $R(x) = f(x) - p(x)$ 称为插值余项，它标志着插值精度。当估值点 $\bar{x}$ 属于包含结点 $x_0, x_1, \cdots, x_n$ 的最小闭区间时，称相应的插值为内插，否则称为外插。

插值是观测数据处理和函数制表所常用的方法，又是数值积分、数值微分、微分方程数值解、非线性方程求根、曲线或曲面拟合的依据。

常用以下插值方法。

多项式插值　插值函数类取作代数多项式类的情形，是最常用的一种插值。此时对 $[a,b]$ 上的任何实值函数 f 都有唯一的次数不超过 n 的多项式函数 p 满足插值条件 (1)，p 称为 f 的插值多项式函数。当 f 在 $[a,b]$ 上 $n+1$ 次可微时，插值余项为

$$R(x) = f(x) - p(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}\prod_{i=0}^{n}(x - x_i)$$

其中 ξ 是包含 $x, x_0, x_1, \cdots, x_n$ 的最小闭区间中的某一点。适当选择结点 $x_0, x_1, \cdots, x_n$，可得到较好的插值效果。特别地，当结点 x_k 取作切比雪夫多项式的零点，即

$$x_k = \cos\frac{2k+1}{2(n+1)}\pi, \quad k = 0,1,\cdots,n$$

时，在一定的意义下，是插值结点的最佳分布。

常见的两种多项式插值公式如下：

(1) 拉格朗日插值公式

$$p(x) = \sum_{i=0}^{n} f(x_i) l_i(x)$$

其中

$$l_i(x) = \prod_{\substack{j=0\\ j\neq i}}^{n}\frac{x - x_j}{x_i - x_j}, \quad i = 0,1,\cdots,n \qquad (2)$$

称为拉格朗日插值公式的基函数。它们具有性质

$$l_i(x_j) = \begin{cases} 1 & j = i \\ 0 & j \neq i \end{cases}$$

(2) 牛顿插值公式

$$p(x) = f(x_0) + f(x_0, x_1)(x - x_0) + \cdots + f(x_0, x_1, \cdots, x_n)(x - x_0)(x - x_1)\cdots(x - x_{n-1})$$

式中，$f(x_0, x_1, \cdots, x_k)$ 为函数 f 在点 $x_0, x_1, \cdots, x_k$ 上的 k 阶差商。各阶差商由下列递推公式确定：

$$f(x_0, x_1) = \frac{f(x_0) - f(x_1)}{x_0 - x_1}$$

$$f(x_0, x_1, x_2) = \frac{f(x_0, x_1) - f(x_1, x_2)}{x_0 - x_2}$$

$$\vdots$$

$$f(x_0,x_1,\cdots,x_n)=\frac{f(x_0,x_1,\cdots,x_{n-1})-f(x_1,x_2,\cdots,x_n)}{x_0-x_n}$$

拉格朗日插值公式和牛顿插值公式是同一插值多项式$p(x)$的不同表现形式，因此它们有相同的插值余项。但在实际计算中，后者较为方便，若要增加新的插值结点，只需相应地添加新的项即可。

埃尔米特插值 带有导数值的插值问题。即插值条件为

$$H(x_i)=f(x_i),\quad H'(x_i)=f'(x_i),\quad i=0,1,\cdots,n \tag{3}$$

在所有次数不超过$2n+1$的多项式中，满足插值条件(3)的多项式是存在且唯一的，并可表示为

$$H(x)=\sum_{i=0}^{n}\{f(x_i)[1-2(x-x_i)l_i^1(x_i)]l_i^2(x)+f'(x_i)(x-x_i)l_i^2(x)\} \tag{4}$$

式中，$l_i(x)$为拉格朗日插值基函数。H称为函数f的埃尔米特插值多项式函数。当f在$[a,b]$上是$2n+2$次可微时，插值余项为

$$R(x)=f(x)-H(x)=\frac{f^{(2n+2)}(\xi)}{(2n+2)!}(x-x_0)^2(x-x_1)^2\cdots(x-x_n)^2$$

式中ξ是包含$x,x_0,x_1,\cdots,x_n$的最小闭区间中的某一点。由于埃尔米特插值多项式在结点处与被插值函数取值相同，且变化率也相同，因此，它通常比拉格朗日插值多项式能更好地近似被插函数。式(4)是一种最基本、最重要的埃尔米特插值多项式。此外，在插值结点上，作为插值条件，还可以给出逐次高阶微商值，并且在每个结点上的插值条件个数也可以是互不相同的。

分段插值 在被插值函数不够光滑或插值结点选择不当时，高次插值多项式常常在被插值函数附近激烈地摆动，不能逼近被插函数。其次，高次插值多项式常常将插值条件的数据中含有的误差过分地放大和扩散。因此，实际上很少采用高次多项式插值，而是先将全区间分成许多小区间，然后在每个小区间上采用低次插值(一次、二次或三次插值)。这样的方法称为分段插值法。实践表明，用分段低次插值多项式逼近被插值函数往往比在全区间上用高次插值多项式逼近效果更好。

样条插值 它是一种在相邻插值曲线段的连接处具有一定光滑度的分段插值。最常用的是三次多项式样条插值：给定结点$x_0<x_1<\cdots<x_n$和相应的函数值$f(x_0),f(x_1),\cdots,f(x_n)$，取内结点$x_1,x_2,\cdots,x_{n-1}$作为分段点，寻求一个插值函数$S(x)$，它在每个小区间$[x_{i-1},x_i]$$(i=1,2,\cdots,n)$上分别都是三次多项式，在结点处满足插值条件

$$S(x_i)=f(x_i),\quad i=0,1,\cdots,n$$

并在每个分段点处满足直到二阶微商的连续性条件

$$S(x_i-0)=S(x_i+0)$$
$$S'(x_i-0)=S'(x_i+0)$$
$$S''(x_i-0)=S''(x_i+0)$$
$$i=1,2,\cdots,n-1$$

这里共有$4n-2$个条件，而分段三次多项式$S(x)$在每个小区间上均含有4个系数，共计$4n$个待定系数。因此，在端点x_0和x_n处分别给定一个边界条件之后，插值问题有唯一解。更一般地，若插值函数$S(x)$为分段k次多项式，在诸结点上取给定值，并且在各分段点处有直到$k-1$阶的连续微商，则$S(x)$称为k次多项式样条插值。

三角插值 插值函数类为三角多项式函数类的情形。当被插函数f是以2π为周期的函数时，通常用n阶三角多项式

$$T(x)=a_0+\sum_{k=1}^{n}(a_k\cos kx+b_k\sin kx)$$

作为插值函数类。对于任取的$2n+1$个不同的实数$x_0,x_1,\cdots,x_{2n}$，只要它们两两之差都不是2π的整数倍，则满足插值条件

$$T(x_i)=f(x_i),\quad i=0,1,\cdots,2n$$

的阶数不超过n的三角多项式$T(x)$存在并且唯一。同拉格朗日插值公式的结构相似。$T(x)$可表示为

$$T(x)=\sum_{i=0}^{2n}f(x_i)\left(\prod_{\substack{j=0\\j\neq i}}^{2n}\frac{\sin\frac{x-x_j}{2}}{\sin\frac{x_i-x_j}{2}}\right)$$

有理插值 插值函数类取为有理函数类的情形。当被插函数具有极点或其他奇点时，用有理插值往往很有效。假定已知被插函数$f(x)$在$m+n+1$个互不相同的点$x_0,x_1,\cdots,x_{m+n}$处的值为$f(x_0),f(x_1),\cdots,f(x_{m+n})$，有理插值就是求一个形如

$$R_{m,n}(x)=\frac{a_0+a_1x+\cdots+a_mx^m}{b_0+b_1x+\cdots+b_nx^n}=\frac{P_m(x)}{Q_n(x)}$$

的有理分式函数，使其满足插值条件

$$R_{m,n}(x_i)=f(x_i),\quad i=0,1,\cdots,m+n$$

并以$R_{m,n}(x)$作为$f(x)$的近似值。当$R_{m,n}(x)$的分子和分母有公因式时，可约掉公因式，将所得的既约分式和原来的分式看作同一分式。满足上述插值条件的有理分式$R_{m,n}(x)$若存在，则是唯一的。虽然$R_{m,n}(x)$有时可能不存在，但很多情况下，$R_{m,n}(x)$

还是存在的，其系数 a_i 和 b_i 可通过解由插值条件导出的线性方程组

$$Q_n(x_i)f(x_i) = P_m(x_i), \quad i = 0,1,\cdots,m+n$$

得出。

参考文献

Гончаров В Л. 函数插补与逼近理论. 路见可，译. 北京：科学出版社，1958

（沈毅）

chaxun chuli

查询处理（query processing） 数据库管理系统执行查询语言的数据操纵语句的过程。

查询处理大致可以分为以下 4 个阶段。①语法分析 对给定的数据操纵语言进行分析，报告语句中可能出现的语法错误。如果没有语法错误，就转入下步处理。②语义检查 根据有关的模式定义，安全性规则及完整性约束对查询进行语义检查，报告有关错误。如果通过检查，就将原始语句转换成适合于后续处理的内部表示形式。③查询优化 对查询的各种可能的执行方案进行成本评估，选择代价较低的方案作为执行计划。执行计划是查询优化阶段的输出，由实现原始操作的可执行的内部代码组成。④执行 系统执行优化阶段产生的执行计划，回送查询结果。

查询的执行方案又称为查询计划，通常以树形结构表示，其中，叶子节点为对存储的数据的访问，内部节点为查询的原子操作。查询计划又可以细分为逻辑查询计划和物理查询计划。

在逻辑查询计划中，原子操作为查询算子。数据库管理系统通常为同一个查询算子提供多种不同的实现。在物理查询计划中，原子操作为对于某一个查询算子的具体实现。

在物理查询计划中，每个原子操作从子节点获得输入，进行处理后将结果返回给父节点。查询的中间结果，即原子操作的输入与输出，存放在缓存中，由数据库管理系统统一进行管理。在现代数据库管理系统中，原子操作通常以迭代器形式进行描述。在查询的执行阶段，原子操作在数据库线程中执行，由数据库管理系统统一进行调度。

同一个查询可以表示为逻辑上等价的不同逻辑查询计划。对于同一个查询算子，每一种实现所需的时间和空间代价都不一样。因此，一个逻辑查询计划，通过将查询算子替换为不同的实现，可能对应于执行代价不同的多个物理查询计划。查询优化器挑选代价较低的物理查询计划进行执行。

查询优化器分为两个部分：代价估计与查询计划选择。在查询计划选择中，最核心的问题是对于连接操作的排序。在对连接操作排序时，常用基于动态优化的算法，即：首先从查询计划的叶子节点开始，选择代价最低的选择、投影等查询算子实现；然后，枚举所有可能的两输入的连接操作，为每一个连接操作，选择代价最低的连接实现算法；基于上一步的 k 个输入的连接操作结果，枚举所有可能的（$k+1$）输入的连接操作，并为每一种连接组合选择代价最低的实现。迭代以上步骤直至包含了所有需要连接的输入为止。

对于复杂的查询，如包含子查询的查询，基于动态优化的算法本身的计算代价较大。因此，常用基于启发式规则的方法进行查询优化。参见“**查询优化**”。

当查询计划的代价无法在查询执行前进行准确评估，或代价在查询执行过程中可能发生变化时，查询处理时优化与执行两个阶段可以被合并。这种技术被称为适应性查询处理（adaptive query processing）。此时，数据库管理系统根据当前的环境参数和数据，动态调整查询的执行计划进行执行。

参考文献

1. Garcia-Molina H, Ullman J D, Widom J. Database systems—the complete book. 2nd ed. Readings. Pearson Education 2009

2. Hellerstein J M, Stonebraker M, Hamilton J R. Architecture of a database system. Foundations and Trends in Databases, 2007, 1(2): 141-259

3. Deshpande A, Ives Z. G, Raman V. Adaptive query processing. Foundations and Trends in Databases, 2007, 1(1): 1-140

（周傲英 钱卫宁）

chaxun youhua

查询优化（query optimization） 确定对给定查询的高效执行计划的过程。所谓执行计划也称为查询树，它由一系列内部的操作符组成，这些操作符按一定的运算关系构成查询的一个执行方案。高效的执行计划有两方面的衡量标准：一是使中间结果最小化；其次是采用尽可能好的操作算法。因为再好的执行计划，若没有好的操作算法，查询效率同样难以改善。

传统的查询优化，按优化层次或优化内容分为两个部分，即代数优化和非代数优化，也称为逻辑优化和物理优化。前者主要是依据**关系代数**的等价变

换做一些逻辑变换，以求得优化。后者则主要是根据存储路径和排序来优化数据存取。

统计信息 查询优化的基础，通常包括：①一个关系中的元组数；②关系中元组的长度；③关系中在特定属性上值的分布情况；④**索引**等。根据这些统计信息，我们可以估算各种运算结果的大小以及运算的执行代价。例如，在有几个索引可用的情况下，利用统计信息可以有效地判定最佳的存取路径。

查询优化的基本方法 目前比较流行的优化方法是基于代价的方法和启发式方法。

基于代价的方法首先要生成所有可能的候选计划，通过统计信息和存取路径确定每一计划中的各个操作的代价和综合代价，最后选择代价最小的一个。基于代价方法的最大问题是优化代价较大。

为降低优化代价，许多系统使用了启发式规则来减少候选计划。一些早期的系统甚至只采用启发式方法来做优化。启发式方法基于的假定因素太多，优化的代价虽小，但优化的准确程度较差，目前已很少单独使用。常见的启发式规则有：选择投影尽量下移；连接中应尽量让小表先进行连接等。

现在实际的优化器基本都倾向于将两种方法结合起来使用，即利用启发式方法尽量减少候选计划，利用基于代价的方法准确地确定执行计划。只是各个系统在具体做法上稍有不同。例如 System R 用启发式过程产生 SQL 嵌套块，用基于代价的方法分别处理每一个嵌套块。Ingres 使用了启发式的分解来把一个查询分解为一组单表子查询，然后用基于代价的方法生成这些子查询的执行计划。

除以上两种优化方法外，目前还有语义优化方法。

语义优化建立在系统对数据库模式有一定的理解上。对用户提交的查询，优化器可以利用系统掌握的知识来简化查询计划，或省略结果为空的查询计算。语义优化虽然在目前的实际系统中还没有实现，但它对未来的关系数据库系统查询效率的改善将是至关重要的。

传统的优化方法已发展得很完善，但面对数据库应用的不断发展，其局限性也日益明显。首先传统方法基本是针对关系代数，对数据库标准查询语言 SQL 中引入的许多超出关系代数的成分，例如嵌套子查询，无法做相应的优化处理。另外现时的应用环境日益复杂，高效率的应用需求日益迫切，并行计算环境日益普及，因此适应这些新的应用环境和需求的查询优化方法被不断地提出来，从而弥补了原有方法的不足。新方法包括流水线方法，动态优化方法和并行查询的优化方法等。

流水线 传统的查询优化模型都是基于集合运算的，即一个操作的输入和输出都是一个集合，这样在一连串的操作中，就会形成大量的中间临时结果，通常在系统中以临时表的形式存在。过多的临时表会消耗大量的系统资源，特别是在中间结果很大的情况下。流水线的策略正是为了打破这一传统方法的瓶颈，其基本思想是整个查询就像一条生产线，每个操作就像生产线上的装配工，只是生产线上流动的不是零件，而是元组。因此流水线上操作的输入输出不再是集合，而是元组流。流水线方式带来的直接好处是减少生成临时表，降低因临时表增加的 I/O 开销。流水线方式的另一个优点是它特别适合在多线程或多 CPU 下实现，此时流水线上的操作可以在多个线程或处理器上并行地执行，查询效率自然会大大提高。

动态查询优化 流水线策略解决了查询计划内部数据动态流动的问题。而动态查询优化要解决的是查询计划的动态生成问题。传统的查询优化所生成的查询计划基本是静态的。即在查询执行前，根据启发式规则，统计信息和存取路径确定一个优化的查询计划。通常优化的查询计划可以执行多次，但优化只进行一次，以便减少优化的代价。这样的优化称为静态优化。但在一些动态性较强的应用环境下，数据变化量很大，静态优化的结果往往到执行的时候已与实际情况出入很大。动态查询优化方法就是将优化的时间尽量后移，使优化的结果能更真实地反映数据的实际情况。

并行查询优化 在并行环境下，查询优化又面临了许多新的问题。由于并行数据库环境中存在多个处理器，并行查询优化应尽可能地使每个操作并行处理，充分利用系统资源提高并行度来达到提高系统性能的目的，因此它的实现与传统的数据库查询优化有所不同。具体表现在：首先并行查询优化的查询搜索空间过于庞大。并行查询优化是要在所有可能的串行查询计划的并行化方案中选出代价最小的并行查询计划。很显然，并行查询计划的搜索空间要比串行查询计划的搜索空间大得多，因此对它进行穷举式搜索是不实际的。必须采用启发式规则对执行计划进行一定的削减，以减少搜索的空间。其次并行查询优化的未知因素太多。由于并行数据库系统中用户、查询、操作、甚至操作内的多个步骤间都可能并行执行，系统资源必须在并行执行的各

个单元之间进行合理分配，而这种分配往往是在执行时动态进行的，许多影响查询执行代价的参数，包括数据库的内容（如关系属性值的分布），物理模式（如索引类型），系统参数值（如可用的缓冲区数目和空闲 CPU 数目）以及查询变量在编译阶段都是未知的，只有在执行时才能确定。而以前的数据库系统所采用的静态查询优化方式是在编译时通过对这些未知的参数值进行假设，选取一个较可能的值来进行优化。显然，在并行的多用户环境之下，这种方式不能充分满足系统的要求，有必要针对它提出新的查询优化方案。目前已有许多人对这一问题进行了研究，并提出了许多有效的解决办法。

参考文献

1. Silberschatz A, Korth H F, Sudarsham S. Database system concepts. 6th ed. McGraw-Hill, 2011

2. 王珊，萨师煊. 数据库系统概论. 4 版. 北京：高等教育出版社，2006

（周立柱　孟小峰）

chacuo kongzhi

差错控制（error control）　数字信号（数据码元）在通过实际信道传输的过程中，可能产生错误（误码），采用能够检测或纠正误码的编码技术和其他控制技术，恢复原始数据的方法称为差错控制。

传输误码产生的主要原因有：一般信道的噪声干扰、无线传输信道的多径干扰等。光盘和硬盘等介质在存储数字信息时，也会产生误码。

差错控制技术广泛应用于计算机和通信各个领域，其具体实现方法大致有两种：

（1）利用纠错编码由接收方根据接收到的数据自行纠正产生的误码；

（2）利用检错编码在接收方发现差错后，通过指令通知发送方进行重传等操作以消除误码。

因此，差错控制也称为误码检测与误码纠错（error detection and correction）。随着差错控制编码理论的完善和数字电路技术的发展，差错控制编码已经成功地应用于各种通信系统中，如在卫星通信中一般用卷积码或级连码进行前向纠错，而在有线数据通信中一般用分组码进行反馈重传。

差错控制理论和技术也在计算机、磁记录与各种存储器中得到日益广泛的应用。

参考文献

Hamming R W. 编码和信息理论. 朱雪龙，译. 北京：科学出版社，1984

（周杰　张凌）

chanpin shengming zhouqi guanli

产品生命周期管理（product life-cycle management，PLM）　一种支持产品信息在全企业和产品全生命周期内（从概念到生命周期结束）的创建、管理、分发和使用的战略性业务模式和技术。PLM 是在产品数据管理（PDM）基础上发展和扩充而来的。**产品数据管理**（product data management，PDM）是一种管理产品设计相关信息和与产品相关的设计过程的技术。PLM 包含了 PDM 的全部内容，加强了对产品生命周期内跨供给链所有信息的治理和利用。

产品数据管理技术最早出现在 20 世纪 80 年代初期，目的是为了解决大量工程图纸、技术文档的计算机管理问题和**计算机集成制造系统**（CIMS）中**计算机辅助设计**（CAD）、**计算机辅助制造**（CAM）、**计算机辅助工艺规划**（CAPP）等单元技术间“信息孤岛”的集成问题。20 世纪 90 年代，随着经济全球化和企业信息化发展，产品制造商开始了协同产品生命过程管理，对产品全生命周期信息、过程和资源进行管理。产品数据管理 PDM 开始逐渐向产品生命周期管理 PLM 演变，其重点逐渐转向支持生命周期中不同应用领域和生命周期阶段的集成和协作。21 世纪初，随着互联网技术的深入发展，PLM 领域出现了不少基于互联网的协作型 PLM 产品。近年来，随着企业向集团化发展，一些大型 PDM/PLM 系统开始注重支持异地产品协同设计。这些新型 PLM 系统致力于建立一个能够满足产品生命周期过程中不同领域和开发阶段信息管理与协调的整体解决方案，能够协同和管理产品设计、开发、制造、销售及售后服务等信息集成，实现真正意义上的产品生命周期管理。

典型的 PDM/PLM 体系结构由 4 层组成：①用户界面层　是系统与用户交互的媒介；②功能层　实现 PDM/PLM 的核心功能；③数据层　实现对象模型管理及数据的处理与交换；④支撑层　包括计算及硬件环境、操作系统、网络通信协议及数据库等支撑环境。PDM 明确定位为面向制造企业，以产品为管理的核心，以数据、过程和资源为管理信息的三大要素。PDM 的基本工作原理是，在逻辑上将各个 CAX 信息化孤岛集成起来，利用计算机系统控制整个产品的开发设计过程，通过逐步建立虚拟的产品模型，最终形成完整的产品描述、生产过程描述以及生产过程控制数据。PDM 进行信息管理的两条主线是静态的产品结构和动态的产品设计流程，

所有的信息组织和资源管理都是围绕产品设计展开的。技术信息系统和管理信息系统的有机集成构成了支持整个产品形成过程的信息系统。从总体上看,PDM 提供给用户的主要功能有:①数据和文档管理　管理的对象是各种文档、产品数据以及相应的属性和版本信息。②过程和工作流管理　控制 PDM 操作人员产生和修改数据的方法,为企业提供建立流程和有效执行任务的框架,以提高企业内部的协同和并行工作水平。③产品结构与配置管理　提供在产品的整个生存周期建立和管理产品定义和结构的能力。④零部件分类库管理　管理类型繁杂、数量巨大的零件数据,充分利用标准件、通用件等现有信息。⑤计划与项目管理　企业的项目信息管理,涉及项目任务和期限的指派、项目资源分配、人员组织结构、人员角色分类、用户信息库、瓶颈分析以及触发提醒等内容。相对于 PDM 而言,PLM 是以产品的整个生命周期过程为主线,在 PDM 的基础上融合协同产品工具,并有效集成 CAX、ERP、供应链管理(SCM)、客户关系管理(CRM)、物料清单(BOM)和 CAPP 等应用系统,从而成为支持企业运行的企业信息化管理平台。PLM 提供一整套的业务解决方案,把整个企业的人员、过程和信息有效集成在一起,遍历产品从概念到报废的全生命周期,支持与产品相关的协作研发、管理、分发和使用产品定义信息,从而简化工作流程,缩短新产品从创意到上市的时间。采用基于互联网环境的 PLM 系统的主要功能除了 PDM 相关功能外,还包括:①产品组合管理　从战略、市场、技术、风险和现金流等角度对企业处于不同生命周期的产品组合进行分析,对产品规划、产品开发和产品退出等决策进行支持。②采购和外包管理　以整个产品生命周期为视角,通过企业战略采购与产品设计部门通力配合来降低产品的生产成本。③用户需求管理　对已有和未来产品的顾客需求实施管理。④协同产品设计　为处于不同部门、企业、地域的产品开发团队提供协同设计的环境和工具。⑤系统集成管理　通过接口与 CAX、CAPP、BOM 和 ERP 等系统集成,形成统一的数据信息模型,对整个产品生命周期内所需的各种数据进行管理。

PDM/PLM 应用领域已经遍布航空航天、汽车及交通运输、工业机械、重型装备、造船及船舶、能源、高科技电子、消费产品、零售业等行业。PLM 还逐渐向服装服饰及制鞋、快速消费品、生命科学与制药,乃至钢铁、化工等流程行业以及建筑业延伸。

PDM 的主要发展方向为:①采用 Web、网络、分布式计算等新技术;②与企业资源计划(ERP)等其他信息系统的功能渗透;③向整体集成解决方案发展;④向全企业级方向发展。PLM 的主要发展趋势为:①可定制化的解决方案　为特定企业提供定制的数据模型、业务模型及解决方案,实现快速、安全、稳定并且低成本的部署。②高效多层次协同应用　覆盖从产品市场需求直到产品报废回收等全过程的管理,并解决产品不同阶段、不同参与人员和组织之间的协同。③多周期产品数据管理　从单一产品生命周期管理向多周期产品数据管理发展,支持重复迭代的产品更新换代的多生命周期管理。④知识共享与重用管理　实现对于企业长期积累的数据和知识的管理、共享与传播。⑤数字化仿真普及应用。涉及产品生产制造的仿真和管理过程的仿真等。

参考文献

1. 约翰-斯达克, 祁国宁. 产品生命周期管理: 21 世纪企业制胜之道. 北京: 机械工业出版社,2008

2. 邓超. 产品数据管理(PDM)规范应用指南. 北京: 中国经济出版社,2007

3. 李善平,刘乃若,郭鸣. 产品数据标准与 PDM. 北京: 清华大学出版社,2002　(钟诗胜)

chanpin shuju jiaohuan biaozhun

产品数据交换标准(product data exchange standards)　研究完整的产品模型数据交换的标准化技术。它支持在产品生命周期内对产品数据进行完整一致的描述与数据交换。它以通用的数据交换标准提高产品设计制造过程乃至产品生命周期内数据交换的效率,保证产品数据传输的完整、可靠和有效。它是计算机集成制造(CIM)、**计算机辅助设计**(CAD)/**计算机辅助工程**(CAE)/**计算机辅助工艺规划**(CAPP)/**计算机辅助制造**(CAM)集成和**产品数据管理**(PDM)的关键技术之一。

对产品数据交换标准技术产生最大影响的当属国际标准 STEP 的工作。1983 年 12 月,国际标准化组织成立了工业自动化系统技术委员会,下设工业数据分技术委员会,即 ISO/TC184/SC4,专门研究产品数据交换标准。1988 年底,OSI/TC184/SC4 发布了产品数据表达与交换的国际标准(standard for the exchange of product model data, STEP),其 ISO 的正式代号为 ISO 10303。STEP 作为一个国际标准受到了全世界的高度重视。此后,STEP 标准就成为全世

界通用的产品数据交换标准。

STEP 标准涵盖一个中性格式的计算机化的产品模型。该模型包含整个产品生命周期内从设计到分析、制造、质量控制、检查和产品服务功能的所有信息,如几何形状、拓扑信息、形位公差、表面粗糙度、材料特性、工艺特性、设计特性、功能特性、装配结构、有限元分析等信息,并对产品数据交换进行了描述,在产品的全生命周期内保证了产品信息的完整性。

STEP 标准的体系结构分为应用层、逻辑层和物理层三个层次。①应用层,是面向具体应用的一个层次,包括应用协议及对象的抽象测试集。②逻辑层,是一个与具体实现无关的层次,包括集成通用资源和集成应用资源及由这些资源建造的一个完整的产品信息模型。它是从实际应用中抽象出来的,总结了不同应用领域中信息的相似性,使 STEP 标准的不同应用间具有可重用性并使数据冗余最小化。③物理层,是指具体实现方法的一个层次,它给出具体在计算机上的实现形式。

STEP 标准不仅是一项标准,而且是一组标准的总称,由五大部分组成,即 STEP 标准的描述方法、集成资源、应用协议、实现形式、一致性测试等;它们可划分为两部分:①STEP 标准数据模型,包括通用集成资源、应用集成资源、应用协议;②STEP 工具,包括描述方法、实现方法、一致性测试方法和抽象测试套件。STEP 标准的基本原理和主要的二维和三维产品建模应用协议已经成为正式的国际标准,它不仅适合于交换文件,也适合于作为执行和分享产品数据库和存档的基础。基于 STEP 标准的产品数据定义、建模、传输与交换的软件开发和产业化应用,对于解决制造业信息化环境下的设计和制造(CAD/CAM)数据交换和企业数据共享问题具有很大的实际意义。

产品数据交换标准的应用领域很广,可应用于机械、电子、航空航天、汽车、船舶等多个制造行业领域。典型的应用场合有:①产品开发过程中的应用,如设计部门内群体合作、产品全生命周期设计、集成化产品开发、分布及并行作业、产品数据管理与长期存档等;②产品设计制造与生产管理过程中的产品信息交换与集成,如在计算机集成制造系统中 CAD、CAPP、CAM、CAQ、ERP 之间的数据交换等;③计算机辅助应用系统供应商之间接口的标准化和产品概念模型的标准化等。

产品数据交换标准技术的主要发展趋势表现在行业与产品级的数据表达与交换标准、复杂产品建模技术、基于产品数据交换标准的产品全生命周期管理与信息集成、产品数据交换标准对于网络化制造与虚拟制造的支撑技术等方面。

参考文献

1. 中国标准出版社. 工业自动化系统与集成 产品数据表达与交换 第 1 部分: 概述与基本原理. 北京: 中国标准出版社,2008

2. 红军,张怀存. STEP 标准及其企业信息化. 机械加工与自动化,2003

(钟诗胜)

chanshengshi biaoshi

产生式表示(production representation) 描述一些事件的存在导致另一些事件产生的一种**知识表示**方法。用符号方法表述如下: if A then B,或 A→B,其中 A 称为前件;B 称为后件;→表示由 A 为真导致 B 为真。产生式表示是最常用的一种知识表示,许多**专家系统**采用产生式表示来设计。

产生式表示源于 1943 年由 Emil L. Post 提出的 Post 系统。在这个系统中定义了符号串替换规则,其计算能力等价于图灵机。1965 年 Allen Newell 和 Hilbert A. Simon 将心理学中的“刺激-反应”概念与产生式表示方法联系起来,用来建立认知模型,描述事物的因果关系。E. A. 费根鲍姆(Edward A. Feigenbaum)于 1965 年开始研制的 DANDRAL 专家系统,采用了产生式表示,DANDRAL 被公认为第一个专家系统。

以 Post 系统为例来说,如果 a 和 b 为字母,可用这些字母组成有限符号行。如果有规则:

$$a \to ab,$$

则有下列关系: abab→abbab→abbabb→abbbabb

当描述事实时,前件可以省略,如:

→孔子是中国人

为了表示各种知识,对前件、后件的符号串给予不同定义,赋予不同意义。例如:允许前件有多个项,每项是字符串、带变量的字符串、三元组(对象、属性、值)、命题、一阶谓词、数学公式等。允许用逻辑连结词∧和∨组合多个项。后件一般有一项,表示结论或操作。

在实际应用中,产生式表示中的项还可以表示图形、图像、数组、矩阵、分子结构等。产生式表示的范围非常广泛,举例如下:

if　(动物有翅膀)∧
　　(动物会飞)∧
　　(动物有两条细腿)

then 动物是鸟 CF = 0.85

其中,CF 表示可信度。

又例如,在 MYCIN 系统(一个医疗诊断专家系统)中,用三元组形式表示:

if (细菌,染色斑,革兰氏阳性)∧
(细菌,形态,球状)∧
(细菌,生长结构,链形)
then (细菌,类别,链球菌)

在解决问题中,一个论域需要用一组产生式规则才能完整地刻画因果关系,在产生式系统中所使用的符号或符号串不能发生矛盾。用符号串表示概念、事物;用产生式规则表示事实、规律、经验和公理等。

一个产生式系统由三部分组成:一组产生式、一个综合数据库和一个控制系统(亦称推理机)。

综合数据库包含所要解决问题的初始数据、目标数据,以及当产生式执行后可能改变用来描述中间运行状态的数据。

控制系统的运行表现为激活一串规则,并执行这些规则。每一条规则的激活条件由综合数据库决定。执行规则后会改变综合数据库,并决定下一条规则的激活条件,直到满足结束条件(推出目标数据或找不到可被激活的下一条规则)为止。这个过程称为推理,其步骤具体描述如下:

① 将初始数据送到综合数据库;

② 由综合数据库决定激活规则的条件;

③ 选出所有符合条件的规则;

④ 根据冲突消解方法,选择其中的一条规则,并激活这条规则;

⑤ 执行这条规则,同时可能改变综合数据库;

⑥ 检查当前综合数据库是否与目标数据一致?如一致,则停机;如不一致,则执行②。

上述产生式推理是由综合数据库控制的。还有一种情况是由规则本身来控制激活条件的,这时,上述推理步骤在激活条件处要加以改动,其余步骤基本相同。

当具有多个推理部件时,可以考虑并行推理,并行执行不相关的推理步骤,提高运算效率。因推理机制是树型结构的,越靠近树根结构,推理的并行粒度越低,并行效率也就越低。

Victor R. Lesser 提出了一种称为黑板结构的产生式系统方法,它把综合数据库推广为黑板;把一条产生式规则推广为一个知识源,允许由一组产生式或其他知识表示组成;将激活产生式条件,推广为激活知识源的条件。黑板结构可以解决复杂的专家系统控制问题。

在产生式系统中,有一种用来描述如何使用规则的产生式规则,即所谓规则的规则。例如,在冲突消解中将所有符合激活条件的规则进行排序。又如,描述规定激活规则的条件。在下面 MYCIN 系统中所使用的激活规则就是这种情况:

if ①(感染是骨盆脓肿)∧
②(前件涉及肠杆菌的规则)∧
③(前件涉及细菌是革兰氏阳性、形态是杆状的规则)
then 先考虑涉及②的规则,后考虑涉及③的规则 CF = 0.4

当有多个产生式均满足激活条件时,要选择其中一条规则激活并执行,解决这个问题称为冲突消解。冲突消解属于控制系统的任务。针对不同情况,有多种解决方法,例如:按匹配成功的自然顺序;按综合数据库某些数据更新的记录;按运行速度;按事先安排好的优先级;按事先制定的原则等。解决冲突消解是否合理,直接影响问题求解的效能。

各国都很重视产生式表示系统的研究与开发,我国也做了大量富有成效的工作。下面举出几个有代表性的采用产生式表示的系统:

(1)DENDRAL 是有机化合物质谱谱图解析专家系统,由 Stanford 大学的费根鲍姆研制,创建了专家系统的设计原理和方法,产生了重要影响。

(2)MYCIN 为细菌感染疾病医疗诊断专家系统,首先采用具有可信度的产生式表示,提出一套推理公式,由美国 Edward H. Shortliffe 研制。

(3)PROSPECTOR 是由 Stanford 研究所的 Richard O. Duda 等研制的金属矿藏勘探专家系统,在预测钼矿中,获得较大的经济效益,达到了人类专家的水平。

产生式表示的应用领域有:医学、化学、数学、信息处理、交通管理、地质勘探、石油、化工、排水、环境、农业、教育、军事、工程设计、商业、金融、管理、法律等。

产生式表示的优点有:①模块性,每个产生式是基本的知识单元,产生式之间的联结是松散的,与推理机相对独立,增加、删除产生式很方便。模块性成为建立规则库的基本论点;②易维护、易管理;③清晰、直观,接近人类思维方式,容易被人们接受,容易实现人机对话。

产生式表示也存在一些不足:知识获取是一个

普遍的问题。一些领域专家很难总结出规则知识,有时由于知识所限、经验不足,或由于规律与解决具体问题混在一起,使得提炼产生式表示规则工作量很大;还有一致性问题,目前尚难实现自动检查大型产生式表示系统的一致性,由于可能隐含不一致,因而可能导致错误的结果;另外,基于综合数据库的控制不够直观,调试产生式表示系统有困难。

产生式表示是一种用途广泛的知识表示,已经开展了一些有意义的研究课题,例如:具有可信度的产生式推理及可信度的传播;具有模糊谓词的模糊产生式系统;用于控制的实时产生式系统;实用的、开放的产生式系统开发工具;分布式产生式系统;超大规模产生式系统等。

参考文献

1. Hayes-Roth F, Waterman D A, Lenat D B. Building expert systems. Reading, MA: Addison-Wesley, 1983

2. 林尧瑞, 张钹, 石纯一. 专家系统原理与实践. 北京: 清华大学出版社, 1988

3. 王树林, 袁志宏. 专家系统设计原理. 北京: 科学出版社, 1991

4. 王树林. 知识处理论. 北京: 科学出版社, 2009

(王树林)

changliang

常量(constant) 可在编译时刻确定其值,且一经确定便不能更新的计算对象。

常量可在程序内部创建并使用。可借助常量定义来定义常量,以确定其值,常量定义的作用是将一标识符绑定于一值。该标识符即为该常量的名。例如,在 **PASCAL** 语言中,常量定义具有如下形式,即 **const** I = E,其中 I 为标识符,E 为可在编译时刻确定其值的表达式。当编译程序处理到该常量定义时,以 I 为名的常量便具有当时 E 之值,此值一经确定,在相应程序中便不能改变。常量定义的作用既有利于避免书写错误,又可节约存储空间。

参考文献

Watt D A. Programming language concepts and paradigms. Prentice Hall, 1990

(徐家福)

changweifen fangcheng shuzhi jiefa

常微分方程数值解法(numerical solution of ordinary differential equations) 根据给定的定解条件,研究如何采用有效的数值方法将微分方程离散化以及求离散化方程的近似解的方法和过程。按定解条件的不同,可分为常微分方程初值问题和边值问题两类。

一阶常微分方程的初值问题是求函数 $y(x)$,使满足条件

$$\left.\begin{aligned} &y'(x) = f(x, y(x)), \quad a < x < b \\ &y(a) = y_a \end{aligned}\right\} \tag{1}$$

数值解法的基本思想是:先取自变量 x 的一系列离散点

$$a = x_0 < x_1 < \cdots < x_{N-1} < x_N = b$$

对区间 $[a,b]$ 作剖分,将微分方程离散化,求出离散问题的数值解,并将其作为微分方程问题的解 $y(x)$ 的近似。例如,取步长 $h > 0$,令 $x_n = a + nh$,把微分方程离散化成差分方程。利用初始条件解此差分方程,得到解 $y_n, n = 0,1,\cdots,N$。将 y_n 作为 $y(x_n)$ 的近似值,量 $\varepsilon_n = y(x_n) - y_n$ 就是近似解的误差,称为全局误差。数值解法的基本内容包括设计各种离散化方法,求出相应差分方程的解,估计解的误差并研究数值方法的收敛性和数值稳定性等问题。

常用的离散化方法有三种:

(1) 基于数值微分的方法 将方程(1)右端的导数 $y'(x)$ 用某个数值微分公式代替,得到相应的差分方程。例如,在 x_n 点用 $(y_{n+1} - y_n)/h$ 代替 y'_n,得到欧拉向前公式

$$\left.\begin{aligned} &y_{n+1} = y_n + h f(x_n, y_n), \quad n = 0,1,\cdots,N-1 \\ &y_0 = y_a \end{aligned}\right\} \tag{2}$$

同样,在 x_{n+1} 点以 $(y_{n+1} - y_n)/h$ 代替 y'_{n+1},得到欧拉向后公式

$$y_{n+1} = y_n + h f(x_{n+1}, y_{n+1}) \tag{2$'$}$$

(2) 基于数值积分的方法 在区间 $[x_n - ih, x_n + jh]$ 上对问题(1)中的微分方程进行积分,得到公式

$$y(x_n + jh) = y(x_n - ih) + \int_{x_n - ih}^{x_n + jh} f(x, y(x))\,\mathrm{d}x \tag{3}$$

式中的被积函数是 x 的函数。若利用某些结点上的 f 值构造此被积函数的近似函数,并且求出此积分,便得到各种差分方程。特别地,若取 $i = 0, j = 1$,并用 f 在结点 $x_n, x_{n-1}, x_{n-2}, \cdots$ 上的值构造的插值函数代替式(3)中的被积函数,就得到亚当斯外推公式。四阶亚当斯外推公式为

$$y_{n+1} = y_n + \frac{h}{24}(55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3}) \tag{4}$$

式中，$f_n = f(x_n, y_n)$。

（3）基于泰勒展开的方法　在设计一个计算 y_{n+1} 的算法时，在算法的公式中安排一些待定的常数。在函数光滑的假定下，将公式进行泰勒展开，并与解 $y(x_n + h)$ 的相应展开式中 h 的同幂次项相比较，按照要求的精度阶得到待定常数应满足的一些方程。求解这些方程确定待定常数，即可得到算法的公式。由此法可导出**龙格–库塔公式**。

龙格–库塔公式的一般形式为

$$y_{n+1} = y_n + \sum_{i=1}^{k} b_i K_i \tag{5}$$

式中，

$$K_i = h f(x_n + c_i h, y_n + \sum_{j=1}^{k} a_{ij} K_j),\ i = 1,2,\cdots,k;$$

a_{ij}, c_i, b_i 为待定常数。最常用的显式四阶龙格–库塔公式为

$$y_{n+1} = y_n + \frac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4) \tag{6}$$

式中，

$$K_1 = h f(x_n, y_n)$$

$$K_2 = h f\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}K_1\right)$$

$$K_3 = h f\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}K_2\right)$$

$$K_4 = h f(x_n + h, y_n + K_3)$$

按照计算 y_{n+1} 时所用结点上值的多少可以将算法分为**单步法**和**多步法**。单步法是指已知结点 x_n 上的值 y_n 便可计算出 y_{n+1} 值的方法。多步法是指已知结点 $x_n, x_{n-1}, \cdots, x_{n-k+1}$ 的值 $y_n, y_{n-1}, \cdots, y_{n-k+1}$ $(k \geqslant 2)$，才能计算出 y_{n+1} 值的方法，这也称为 k 步方法。对于多步法，给出 y_0 后，需要由其他算法计算出 $y_1, y_2, \cdots, y_{k-1}$ 后（起步），才能使用多步法逐步计算 $y_k, y_{k+1}, \cdots$ 的值。

为了使构造的数值解法具有实用价值，需要研究算法的相容性、收敛性、误差估计和数值稳定性等问题。

相容性　将微分方程离散化成差分方程，并将微分方程的解代入该方程，出现的误差称为局部截断误差。当 $h \to 0$ 时，局部截断误差趋于零，则称该差分方程与微分方程具有相容性。这表示差分方程是微分方程的一种近似。若局部截断误差对 h 的展开式的首项为 ch^{p+1}，其中 c 为常数，则称局部截断误差的阶为 $p+1$，而称相应的算法是 p 阶的。p 越大，表示离散化后的方程与原微分方程的近似精度越高。

收敛性　指当 $h \to 0$ 时，全局误差 $\varepsilon_n \to 0$，即离散问题的解 y_n 收敛于微分方程的解 $y(x)$。这是由数值方法得到的近似解可用的理论基础。对于 p 阶算法，当 $h \to 0$ 时，误差 ε_n 将以 h^p 的速度收敛到零。p 为误差主项中步长 h 的幂次。

误差估计　应用数值方法时最关心的问题是全局误差 ε_n 的估计。理论估计通常只能给出 $h \to 0$ 时的误差阶。一般采用事后估计法，即根据计算过程中获得的关于解的信息进行误差估计。例如通过对不同步长的计算，采用理查森外推法估计误差。利用得到的误差估计便可对数值解进行校正。

数值稳定性　指在计算过程中，某一步上产生的误差一步一步传递下去不会影响数值解的精度。在研究数值稳定性时，微分方程

$$y' = \lambda y \tag{7}$$

常常作为研究微分方程的代表，其中 $\lambda = \alpha + \mathrm{i}\beta$，$\alpha < 0, \mathrm{i} = \sqrt{-1}$。许多数值稳定性的定义都以这个方程为基础给出的。在研究常微分方程数值稳定性时，首先需要研究稳定区域，这可参见文献[1]。

对于常微分方程组初值问题

$$\boldsymbol{y}' = \boldsymbol{f}(x, \boldsymbol{y}),\quad \boldsymbol{y}(0) = \boldsymbol{g}$$

式中，$\boldsymbol{y} = (y_1, y_2, \cdots, y_m)^{\mathrm{T}}$，$\boldsymbol{f} = (f_1, f_2, \cdots, f_m)^{\mathrm{T}}$，$\boldsymbol{g} = (g_1, g_2, \cdots, g_m)^{\mathrm{T}}$ 都是 m 维向量，若微分方程组右端函数 $\boldsymbol{f}(x, \boldsymbol{y})$ 的雅可比矩阵 $\dfrac{\partial \boldsymbol{f}(x, \boldsymbol{y})}{\partial \boldsymbol{y}}$ 的特征值 $\lambda_j = \alpha_j + \mathrm{i}\beta_j,\quad j = 1,2,\cdots,m$，满足 $\alpha_j < 0$，并且比值

$$S = \frac{\max|\mathrm{Re}\lambda_j|}{\min|\mathrm{Re}\lambda_j|} \gg 1$$

称这类问题为刚性问题。对刚性问题应用通常的显式方法[1]时，由于数值稳定性，$|h\lambda_j|$ $(j = 1,2,\cdots,m)$ 受到限制，h 必须取得很小，因而计算量非常大。向后公式、梯形公式都是数值求解刚性问题的有效方法。

各种实际问题导出的不同类型常微分方程边值问题中，较简单的是两点边值问题：求函数 $\boldsymbol{y}(x) = (y_1(x), y_2(x), \cdots, y_m(x))^{\mathrm{T}}$，$x \in [a, b]$，使其满足微分方程组

$$\boldsymbol{y}'(x) = \boldsymbol{f}(x, \boldsymbol{y}) \tag{8}$$

和边值条件

$$\boldsymbol{g}(a, \boldsymbol{y}(a), b, \boldsymbol{y}(b)) = 0 \tag{9}$$

式中 m 维向量函数 $\boldsymbol{f}$ 和 $\boldsymbol{g}$ 都是已知函数。条件（9）

是在两个端点上对解的约束条件。有些实际问题要求解在区间 $[a,b]$ 的多个点上的值满足预定的条件,称这样的问题为多点边值问题。

解常微分方程边值问题的数值解法有差分法和打靶法。

差分法 直接将微分方程离散化,加上边值条件构成一个代数方程组,解此代数方程组即可得到边值问题的数值解。

打靶法 其基本思想是将边值问题化成一系列初值问题求解,适当选择和调整初值 $\boldsymbol{y}(a)$, 使得相应的解 $\boldsymbol{y}(x)$ 满足指定的边值条件。下面以两点边值问题为例进行说明。若已知初值 $\boldsymbol{y}(a)$ 的值,求解初值问题可确定解 $\boldsymbol{y}(x)$, 从而唯一确定 $\boldsymbol{y}(b)$。因此, $\boldsymbol{y}(b)$ 是初值 $\boldsymbol{y}(a)$ 的函数, 记为 $\boldsymbol{y}(b) = \boldsymbol{p}(b,\boldsymbol{y}(a))$。将 $\boldsymbol{p}(b,\boldsymbol{y}(a))$ 代入式 (9) 中的 $\boldsymbol{y}(b)$, 得到只含未知量 $\boldsymbol{y}(a)$ 的代数方程组

$$\boldsymbol{g}(a,\boldsymbol{y}(a),b,\boldsymbol{p}(b,\boldsymbol{y}(a))) = 0 \qquad (10)$$

若将 $\boldsymbol{y}(x)$ 看成是弹道,将 $\boldsymbol{y}(a)$ 看成是射击条件,则 $\boldsymbol{y}(b)$ 就是在射击条件 $\boldsymbol{y}(a)$ 下所达到的靶子。因此打靶法的求解过程是不断地调整射击条件 $\boldsymbol{y}(a)$, 使它和相应的靶子 $\boldsymbol{y}(b)$ 满足预定的边界条件。若 b 固定,从数学上看,打靶法是要求解 $\boldsymbol{y}(a)$ 的方程 (10) 。因此求解非线性方程的各种数值方法都可以应用到常微分方程边值问题。例如,应用牛顿方法,得到牛顿打靶法,其计算步骤如下:

步骤 1 适当选取 $\boldsymbol{y}(a)$ 的迭代初值 $\boldsymbol{y}^{(0)}(a)$, 令 $i = 0$。

步骤 2 数值求解常微分方程初值问题

$$\boldsymbol{y}'(x) = \boldsymbol{f}(x,\boldsymbol{y}(x))$$
$$\boldsymbol{y}(a) = \boldsymbol{y}^{(i)}(a)$$

将得到的解记为 $\boldsymbol{y}^{(i)}(x)$, 特别得到 $\boldsymbol{y}^{(i)}(b) = \boldsymbol{p}(b,\boldsymbol{y}^{i}(a))$。

步骤 3 计算

$$\boldsymbol{g}^{(i)} = \boldsymbol{g}(a,\boldsymbol{y}^{(i)}(a),b,\boldsymbol{y}^{(i)}(b))$$

若满足精度要求,即有 $\|\boldsymbol{g}^{(i)}\| < \delta$, 则停止计算,取 $\boldsymbol{y}^{(i)}(x)$ 为所要求的近似解;否则转步骤 4。δ 为预先给定的精度常数。

步骤 4 计算

$$\boldsymbol{y}^{(i+1)}(a) = \boldsymbol{y}^{(i)}(a) - \boldsymbol{G}_i^{-1}\boldsymbol{g}^{(i)}$$

式中

$$\boldsymbol{G}_i = \left[\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{y}(a)} + \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{y}(b)}\frac{\partial \boldsymbol{p}}{\partial \boldsymbol{y}(a)}\right]\Bigg|_{\boldsymbol{y}(a)=\boldsymbol{y}(i)(a)}$$

步骤 5 令 $i \Leftarrow i + 1$, 转步骤 2。

在实际实现时,可将上述算法作各种变形和简化。

目前,数值求解常微分方程已有许多著名的软件包,如 LSODE 和 RADAU,前者采用亚当斯内插法和向后微分方法,可求解刚性和非刚性问题,运行过程中能自动变阶和变步长,后者采用隐式龙格–库塔方法。

参考文献

1. Gear C W. 常微分方程初值问题的数值解法. 费景高,刘德贵,高永春,译. 北京:科学出版社,1978

2. Hairer E, Norsett S P, Wanner G. Solving ordinary differential equations I, Nonstiff Problems. Berlin: Springer-Verlag, 1987

3. Hairer E, Wanner G. Solving ordinary differential equations Ⅱ, Stiff and differential-algebraic problems. Berlin: Springer-Verlag, 1991

(费景高)

chaodao jicheng dianlu

超导集成电路(superconducting integrated circuit) 将一定数量的超导元件和约瑟夫逊结器件做在一块芯片上以完成一定的电路功能的集成电路。

当一些金属、合金和化合物冷却至低于它们的临界温度 T_c 时,它们的电阻陡然降为零。如果是一个超导回路,则其中的电流将会无限制地流动,除非升高温度或外加磁场,才能中断该超导电性,具有这样性质的物体即称之为**超导体**。迄今为止,超导体可按其 T_c 粗略地划分为两类:一是经典低 T_c 金属性超导体,主要为 Nb(9.5K),NbN(15K),Nb_3Sn(21K),Nb_3Ge(23.2K);另一是高 T_c 氧化物超导体,目前公认的为 $Y_1Ba_2Cu_3O_{7-x}$ (91K), $(Bi,Pb)_2Sr_2Ca_2Cu_3O_{10+y}$(110K),$T_{12}Ba_2Ca_2Cu_3O_{10+\delta}$ (125K)。

若由两个超导体构成一个隧道结,它们之间用绝缘层分开,当此绝缘层减薄到一定程度,且将此结冷却到临界温度以下时,将有电流流过此绝缘层,此即为隧道效应超导电流。它与一般隧道效应不同之处在于不需施加电压,这种超导隧道效应称为约瑟夫逊效应,据此制成**约瑟夫逊结器件**。当电流超过此超导电流的阈值时,器件以极快的速度由零电阻态转变为高电阻态。由于该超导阈值电流是穿透约瑟夫逊结有效面积的函数,故可用施加外磁场来减

小阈值电流，从而使约瑟夫逊结器件由超导态（零电压）转变为正常态（有限电压）。这个电压变化值在3 mV左右（而硅（Si）和砷化镓（GaAs）晶体管的转换电压在77 K时约为200 mV和500 mV），其开关速度可达0.1 ps，工作电流一般低于1 mA，功耗低于1 μW。功耗与开关时间的乘积通常作为比较数字技术的品质因素，约瑟夫逊结的此乘积比目前任何半导体工艺得到的数值小2～3个数量级。但约瑟夫逊结器件没有隔离，没有确定的增益和非逆置性，从而增加了超导集成电路设计的复杂性。超导集成电路的制备与半导体集成电路有许多相似之处，其中制版、光刻、蒸发和溅射工艺基本相同，它无须离子注入、高温扩散和外延淀积工艺，而且约瑟夫逊结器件不用衬底，片子可做得比用半导体工艺的大。目前超导逻辑电路和数字电路较为成熟，已有超导模拟数字转换器、超导移位寄存器和超导动态随机存取存储器。它所用的工作致冷剂为液氦（4.2 K）。上述经典低 T_c 金属性超导体，主要是用铌（Nb）制成的。

利用高 T_c 氧化物超导体制作集成电路的工作已大力开展。目前已研制成高温超导晶体管，它的输入电阻小，输出电阻大，可作为连接固态电子元件和低温超导器件的桥梁，克服了约瑟夫逊结器件由于输出电阻小、输出电压低，很难与半导体器件匹配连接的困难。高温超导晶体管是一种反向场效应晶体管，它的输出电压是由输入电流控制的，在液氮温度（77 K）下工作。可用它制成放大器、振荡器、相移器和混频器等多种电路。

近来提出的一种快单磁通量子（RSFQ）逻辑电路不需要隧道结，而只需用一般的弱连接或高 T_c 超导体与正常金属结（SNS）来组成。它在77 K时的性能优于半导体电路，简单的RSFQ电路其时钟速度已超过100 GHz，期望可达300 GHz以上。另一优点是功耗小，4 K时为100～500 nW/门。在40～50 K时，预期功耗将增加一个数量级，可用于制成各种超高速数字器件，如模数转换器、移位寄存器、数字处理器等。现已利用以 $SrTiO_3$ 为衬底的 $Y_1Ba_2Cu_3O_{7-x}$/Ag/Pb 合金组成的多层薄膜制成邻近效应器件，把三个特性几乎相同的器件用超导线相连构成一个简单的逻辑电路，它具备逻辑加和逻辑乘的功能。RSFQ逻辑电路的近期应用是制造低热耗的模数转换器。

将低 T_c 超导集成电路芯片与半导体集成电路芯片通过高 T_c 超导传输线有效地连接起来组成具有特定功能的超导-半导体混合集成电路，可在不同温度下工作，它将是超导集成电路实用化的发展方向。

参考文献

Hara K. Superconductivity electronics. Prentice-Hall, Inc., 1987

（徐鸿达）

chaodao jisuanji

超导计算机（superconducting computer）

指利用超导约瑟夫森结效应和超导电效应构建的包括数据存储、检索、处理的计算机系统。超导计算机具有运算速度高、功耗小的特点，是理想的非传统高性能计算机系统。

超导电子学是20世纪最为活跃的学科领域之一。自从发现超导约瑟夫森结效应以来，人们一直致力于研究超导数字计算机，期望获得传统半导体技术所难以达到的计算能力。超导计算机根据其对数据的表示和处理机制主要分为基于闩锁结构的高低电位和基于量子干涉器的快单磁通量子两种方式。

早期研究集中于利用超导约瑟夫森结的临界电流高速态转换下所对应的结电压高低电位实现二进制逻辑数字运算，这种由高低电位表示“1”“0”方式与传统半导体数字电路相似。这一时期对超导计算机开展的努力主要是由美国IBM和日本研究计划进行的。然而，这种基于闩锁逻辑机制的超导数字电路最终并未获得成功。导致这类超导数字电路无法获得超过传统半导体数字电路性能的原因主要包括两个方面：①铅合金超导材料的热回复稳定性不能达到实用要求；②闩锁欠阻尼逻辑电路的交流偏置过冲效应限制了运算速度的提高。

1978年A Silver首次提出了采用超导量子干涉器实现以磁量子脉冲方式进行数字运算的概念，并构造了超导磁量子T触发器原型。K Likharev等人于1991年系统提出了一系列快单磁通量子（rapid single flux quantum，RSFQ）数字逻辑电路模型。与此同时，早期超导数字计算机研究所发展起来的铌超导平面电路制备技术为超导超大规模集成电路的实现建立了坚实的基础。在超导磁量子数字电路中，信息是以单磁通量子形式存储在超导量子干涉器环路中，并以光速在超导电路中传输。数字“1”或“0”是以触发量子脉冲到达时电路中有无对应的量子脉冲来表示。由于快单磁通量子脉冲宽度只有 *p* 秒量级，因而可以工作在极高的时钟频率下，实验室中对简单电路测得的最快时钟频率可达

770 GHz。超导单磁通量子数字电路几乎可以实现所有半导体数字电路的基本功能，因而几乎没有人怀疑超导单磁通量子器件能在高性通超级计算机中获得成功。

近年来主要研究是由美国空气动力实验室(JPL)牵头开展的 HTMT(hybrid technology multithreading)计划，以期利用超导快单磁通量子数字电路技术及其他相关前沿技术研制每秒千万亿次浮点运算能力的超导通用超级计算机。

参考文献

1. Hayakawa H, et al. Superconducting digital electronics. Proceedings of the IEEE, invited paper, 2004, 92(10): 1549-1563

2. Likharev K, Semenov V. RSFQ logic/memory family: a new Josephson junction technology for subterherz clock frequency figital systems. IEEE Trans on Appl Supercond, 1991, 1: 3-28

3. Gao Guang et al. Hybrid technology multithreaded architecture. Frontiers of Massively Parallel Computing Symposium, 1996: 98-105

(官伯然)

chaojiedian jiegou

超结点结构(supernode architecture) 大规模**并行处理系统**中支持物理上单一地址空间并由硬件实现**高速缓冲存储器一致性**的**共享存储**多处理机结构。

为了提高可扩展性，大规模并行处理系统多采用分层结构，整个系统被分成不同层次，由不同类型的网络互联，支持不同的存储结构。目前主要有**共享存储**和**消息传递**两类不同的结构层次，超结点对应共享存储结构层次。在一个由多个超结点互连构成的并行系统中，每个超结点拥有单独的共享物理地址空间，超结点间通过消息传递交换数据；超结点内各处理机间拥有单一的物理地址空间并由硬件实现各处理机高速缓冲存储器的一致性，支持各处理机间的数据共享。

超结点结构的主要特点是：①具有由多个处理机，由硬件实现物理上的单一地址空间和高速缓冲存储器一致性；②只有一个操作系统副本；③软件主要采用共享存储编程模式；④任一处理机都能够访问超结点中所有的存储位置和输入输出设备。

超结点结构主要有 3 类，即对称多处理机(SMP)结构、高速缓冲存储器一致性非均匀存储器访问(CC-NUMA)结构和唯高速缓冲存储器存储结构(COMA)。

(1) SMP 结构　在该结构系统中，多个处理机通过高速侦听总线或定制的交叉开关网络与构成单一地址空间的所有存储模块互联，各处理机可直接访问各存储模块的任意位置，且所用存储访问时间相同。因此，SMP 属于集中存储的均匀存储访问(UMA)结构。SMP 中的高速缓冲存储器一致性由**系统总线**硬件通过总线侦听协议实现，对称性体现为各处理机对系统中存储器、外部设备和操作系统服务的访问是对等的。

(2) CC-NUMA 结构　为提高可扩展性，CC-NUMA 结构中存储器是物理分布的，各处理机都有本地存储器并通过系统总线直接相连，一个处理机的本地存储器成为其他处理机的远程存储器并通过定制的实现高速缓冲存储器一致性的**互连网络**访问，各处理机对本地存储器和远程存储器的访问时间不同，访问远程存储器的延迟比访问本地存储器大得多，达一到两个数量级。高速缓冲存储器一致性通过硬件实现的基于高速缓存目录的一致性协议来保持。由于不同存储器访问时间不同，系统软件必须尽可能保证数据的局部性，如采用页面迁移。

(3) COMA 结构　COMA 结构是在 CC-NUMA 结构上的一种演变，它将 CC-NUMA 结构中的所有局部存储器都组织成高速缓存，称为 COMA 高速缓存，因此 COMA 的主存储器是由各 COMA 高速缓存构成的。另一个主要变化是在 COMA 结构中为保证数据局部性而做的数据迁移是由硬件实现的，而不是操作系统。数据迁移的单位不再是页，而是高速缓存的行。

目前广泛采用的超结点结构是 SMP 和 CC-NUMA，最大系统处理机个数分别为 64 和 512，COMA 由于过于复杂而少有真实系统。通过引入分区概念增加操作系统副本使得超结点结构的可用性问题得到一定程度的解决。未来伴随多处理机芯片的问世，芯片本身就可较好地支持直接互连成 SMP 或 CC-NUMA 结构。

参考文献

1. Kai Hwang, Zhiwei Xu. Scalable parallel computing: technology, architecture, programming. McGraw-Hill Companies, Inc., 1998

2. 卢锡城. 关于大规模并行处理机系统可扩展性设计. 中国工程科学，2000

(孟丹)

chaowenben

超文本(hypertext) 一种非线性网状链接结构的文本。由于现代大多数**多媒体文档**都基于超链接结构,这种结构的多媒体文档也常称为**超媒体**,超媒体是超文本的拓展形式。

超文本与以线性形式进行组织的文本有很大不同。它以节点(也称为部件)为单位组织各种信息,一个节点是一个“信息块”,结点内的信息可以是文本、图像、图形、动画、声音或其组合;节点间通过链加以链接,形成一个网状结构。链可以指向文档中的任一部分,也可以指向另一文档。链附属于锚,锚可以是一个字或一个句子。链包含访问目标文档的所有必需的信息,链是可激活的。

超文本中的非线性文档结构思想最早由 V. Bush 在 1945 年提出。“超文本”这个术语由 T. Nelson 于 1965 年发明。当时他开发了一个真正的超文本系统——Xanadu。D. Englebart 是超文本研究的另一先驱,他于 1968 年开发成功交互式多用户超文本系统 NLS。

超文本的最大优点是提供了非线性的信息链接。与普通读物不同,超文本不仅可以顺序地阅读,而且可以选择阅读路径,它提供了自然有效的信息导航。超文本的研究重点是关系管理。超文本中另一个重要的概念是“访问部件”,它指获得部件数据的直观效果,例如文字、图形的视觉效果,声音的听觉效果等。超文本中对于部件访问的重要特征是,访问的时间完全由用户自己决定,触发另一个链接或结束应用程序则终止部件访问。

多媒体是随着计算机对音频和视频的支持技术发展起来的。多媒体主要以演示的形式表现出来,这与基本的超文本有很大的不同。多媒体演示中所包含的部件是按某种规定的次序出现的,因而在多媒体文档中必须引入一个重要的概念——时间。当然,用户仍有控制部件访问或不被访问的能力。但是,正因为有了时间的概念,从而即使在没有用户干预的情况下,演示仍然会发生变化。所以在多媒体文档中,不仅要定义出部件之间的位置关系,而且也需要定义出它们之间的时间关系,即多媒体对象必须包括时空关系。

互联网是超媒体的典型例子。在 20 世纪 90 年代早期 Berners-Lee, CERN 的一位科学家提出应用 world wide web 来满足在 EONR(European Organization for Nuclear Research)与世界各地的研究所和大学中工作的科学家之间以通过简单和直接的方式来满足信息共享的要求。HyperText 是一种链接和访问作为 Web 上节点的各种信息,对这些节点用户可随意浏览。它提供了一种对各种各样类型信息(报告、节点、数据库、计算机文件和在线帮助)单一的用户接口。超媒体是在超文本和多媒体发展到一定阶段的产物。超媒体包括了超文本和多媒体两者的所有特点,也就是说,一方面,它要支持节点、链接等复杂的浏览关系,同时,也要引入多媒体表现,尤其是时间的概念,因为超媒体中的信息最终是以多媒体演示的形式呈现给用户的。作为一个较为严格的超媒体的定义,强调文档中不仅要有多种类型的媒体数据,还要具有时间合成的概念,以体现多媒体的特性。 (徐光祐 潘志庚 史元春)

chaoxietiao luoji

超协调逻辑(paraconsistent logic) 一类非协调但不平凡的逻辑理论。亦译弗协调逻辑、次协调逻辑。对某个命题 A, $\{A, \neg A\}$ 是一个(逻辑)矛盾,一个理论 T 是非协调的,若它至少包含一个矛盾;一个理论 T 是平凡的,若它包含任一命题。超协调逻辑可看成是在非协调(含矛盾)信息情形下推理的形式化。经典逻辑不是超协调的,因为经典逻辑具有平凡性。形式化超协调性的目标就是要破坏经典逻辑中的平凡性,超协调表示可以逃脱经典逻辑的协调性的束缚。超协调一词是 1976 年 Quesada 提出来的。如果以建立第一个完整的逻辑演算作为超协调逻辑的起点,1963 年 da Costa 提出的 Cn 系统是形式化超协调性的开始,而最早研究超协调性则是 1910 年由 Lukasiewicz 的三值逻辑,Vasilev 的意象逻辑(imaginary logic),Jaskowski 的分域逻辑(discursive logic),以及一些相干逻辑(relevant logic)就已经开始了。超协调逻辑与相干逻辑和辩证逻辑(dialethic logic)是密切相关的。追根溯源,超协调概念可以在东西方哲学的各个历史时期找到,它是伴随着对矛盾现象的研究开始的,而矛盾是普遍存在的。现在,超协调逻辑还与一些从不同角度处理矛盾的逻辑推理紧密联系,如**非单调逻辑**等。

超协调逻辑已有众多的逻辑系统。列举三种比较有代表性的超协调逻辑如下。

(1) 超协调命题演算 C_n 是一个分层超协调命题演算系统($1 \leqslant n \leqslant \omega$),具有如下形式特征:①$C_0$, $C_1, \cdots, C_n, \cdots, C_\omega$ 依次严格地减弱演算能力;②经典逻辑 C_0 的所有定理对良构公式在 C_n 中有效,一个公式 A 是 C_n 的良构公式,若 $A^{(n)}$ 是 C_n 的定理,这里

$A^{(n)} = A^1 \wedge \cdots \wedge A^n$,定义 $A^1 = A^0 = \neg(A \wedge \neg A)$, $A^n = (A^{n-1})^0$。形式系统 C_n 通过减弱演绎能力达到形式化超协调性的目的,尽可能地保留经典逻辑,语法简单,由于抛弃无矛盾律,能处理含如悖论这种矛盾的辩证推理,且语义较简单。

(2) 多值逻辑是具有多于两个真值的逻辑系统,经典多值逻辑与经典逻辑一样只指派一个真值(真)定义真理,同样具有平凡性不是超协调的。超协调多值逻辑通过指派至少两个真值定义真理,从语义上可直接具有超协调性。其中,以 1977 年 Belnap 提出的四值逻辑最具代表性。四值逻辑具有四个真值,分别表示经典真 T,经典假 F,矛盾(既真又假)B 和未知(非真非假)N。通过四值二元组形式的双格结构来表示模型,其中第一个分量表示命题具有成真的信息,第二个分量表示命题具有成假的信息,把命题成真和命题成假的信息分别对待使得某个命题可以同时既具有成真的信息又具有成假的信息,从而可包容矛盾具有超协调性。

(3) 准经典逻辑由 Besnard 和 Hunter 于 1995 年提出,定义了两种可满足关系:弱满足关系和强满足关系。前者在结论上用来容忍矛盾而后者在前提上使消解规则成立。在准经典逻辑中,演绎规则使用的次序受到限制,演绎过程可分成两步:第一步,无次序地使用除析取规则之外的演绎规则;第二步,仅使用析取规则。这样,不仅获得超协调性,而且有效地增强超协调逻辑的推理能力。

传统上,超协调逻辑属哲学逻辑范畴,现在计算机科学和**人工智能**中具有重要的应用。使用计算机处理信息,由于信息来源或海量信息自然地包含矛盾,要求信息系统都遵循经典逻辑要求的协调性是不现实的,或是计算上要排除一切矛盾是不可行的,超协调逻辑为更好地解决这类问题提供了强有力的工具。另一方面,在计算机科学和人工智能背景中超超协调逻辑得到进一步的发展。

参考文献

Priest P, Routley R, Norman J, eds. Paraconsistent logic: Essays on the inconsistent. Philosophia Verlag, 1989

(林作铨)

chezai wangluo

车载网络(vehicle Ad Hoc network)　通过提供车辆之间以及车辆与公共设施之间的通信,以最小的延迟发送和接收当前交通状况的警报或者信息的网络。同时由于其拥有 Ad Hoc 网络的一些特性,因此也被称作车载 Ad Hoc 网络(VANET)。

车载网络的目标主要分为以下三点:

(1) 增强道路安全　行驶中的车辆可以通过与其他车辆或路旁公共设施获得一些紧急信息从而避免危险的发生,或者在自己遇到危险的时候通知周围的车辆以避免造成更大的事故。

(2) 提高个人交通出行效率　车载网络可以让车辆通过网络随时随地获得瞬时路况信息,选择更好的路线选择方案,避免交通拥堵,提高出行效率。

(3) 提高生活便利性　车载网络可以为用户提供多种多样的便捷的日常生活服务、娱乐服务等,提高人们的生活质量。

虽然车载网络的发展很迅速,但仍面临许多挑战。首先车载网络在交通安全与效率方面的收益必须得到体现。为了得到车载网络在现实世界的性能,需要在世界范围内进行大量的试验。除此之外,还要提出仿真模型,进一步开发使用平台以及深入研究智能交通管理。特别是在未来的几年中,车辆基础设施一体化工程将高度影响车载网络的发展。

参考文献

Hartenstein H, Laberteaux K P. A tutorial survey on vehicular ad hoc networks. IEEE Communications Magazine, 2008, 46(6): 164-171

(崔勇)

chenjingan

沉浸感(immersion)　能让用户感觉到自己好像完全置身于虚拟世界之中,成为虚拟世界的一部分的技术。用户可以根据自己的需要与虚拟环境中的物体进行交互,从而参与虚拟世界的各种活动,达到身临其境的感觉。

虚拟现实的沉浸感来源于对虚拟世界的多感知性,除了常见的视觉感知和听觉感知外,还有触觉(力觉)感知、嗅觉感知、味觉感知等。另外为了达到综合感知,系统还要提供必要的同步功能,对多种不同的感知信息根据场景的变化或用户动作进行必要的同步,从而使用户在有些时候能同时以多种方式感知环境。从理论上来说,虚拟现实系统应该具备人在现实客观世界中具有的所有感知功能,即实现"看起来像真的""听起来像真的""摸起来像真的""闻起来像真的""尝起来像真的"效果。

沉浸感主要来源于以下几种:

(1) 视觉沉浸　视觉通道给人的视觉系统提供三维立体图形图像显示。视觉沉浸的效果对沉浸感起主要作用。

虚拟现实系统必须向用户提供立体三维效果及较宽视野的场景,类似于我们通常看的立体电影。但在虚拟现实系统中,随着人的运动,所得到的场景也随之实时地改变。一种较理想的视觉沉浸环境是在洞穴式显示设备(CAVE)中,采用多面立体投影系统形成三维投影和交互空间,得到较强的视觉效果。

(2) 听觉沉浸　听觉通道是除视觉外的另一个重要感觉通道,如果加入与视觉同步的声音作为补充,可大大提高虚拟现实系统的沉浸效果。在虚拟现实系统中,主要让用户感觉到的是三维虚拟声音,这与普通立体声有所不同。普通立体声可使人感觉声音来自于某个平面,而三维虚拟声音可使听者能感觉到声音来自于围绕双耳的一个球形中的任何位置。也可以模拟大场景的声音效果,如闪电、雷鸣、波浪声等自然现象的声音。在虚拟世界中,两个物体碰撞时,会发出碰撞的声音,用户根据声音就能准确判断出声源的位置。而碰撞声音的模拟要使用到基于物理的仿真技术。

(3) 触觉(力觉)沉浸

在虚拟现实系统中,可以借助于各种特殊的交互设备,如**传感手套**和力反馈设备,可使用户能体验抓、握等操作的感觉。基于当前科技水平,主要侧重于力觉反馈方面。使用充气式手套,在虚拟世界中与物体相接触时,能产生与真实世界相同的感觉。如用手击球时,不仅能听到拍球发出的"嘭嘭"声,还能感受到球对手的反作用力,即手上感到有一种受压迫的感觉。这里就牵涉到三种感知信息的同步,分别是视觉、听觉和触觉。

(4) 嗅觉沉浸　有关嗅觉模拟的开发是最近几年的一个课题。日本最新开发出一种嗅觉模拟器,只要把虚拟空间中的水果放到鼻尖上一闻,能产生不同气味的装置就会在鼻尖处释放出水果的香味。虽然这些设备还不是很成熟,但对于虚拟现实技术来说,是在嗅觉研究领域的一个突破。

(5) 味觉沉浸　在虚拟现实系统中,除了可以实现以上的各种感觉沉浸外,还有身体的其他感觉如味觉等,但基于当前的科技水平,人们对这些沉浸性的形成的机理还知之较少,有待进一步研究与开发。

参考文献

胡小强. 虚拟现实技术基础与应用. 北京:北京邮电大学出版社, 2009

(潘志庚　胡小强)

chenshuxing biaoshi

陈述性表示(declarative representation)　与**过程性表示**相对应的另一类**知识表示**方法,用于描述客观事物特征及其相互关系。它强调知识的静态方面。陈述性表示是一种表示与运用分开的知识表示方法。陈述性知识被静态地表示为知识片段,需要利用**推理机**来解释和运用,推理机的执行过程对用户不可见。

常见的陈述性表示有**逻辑表示**、**产生式表示**、**语义网络表示**、**框架表示**和脚本表示等。下面给出两个采用陈述性表示方法表达知识的例子。

第一个例子是简单的字母排序知识。采用陈述性表示方法,通过顺序关系"A在B之前,B在C之前,C在D之前,……,Y在Z之前"可以将字母顺序关系显式地表示出来。这种表示很方便:当要判定两个字母的顺序时,通过排序程序对顺序关系进行操作,就可得出结论。

第二个例子是简单的逻辑知识例子。假定有如下事实:

$\forall x(\text{Person}(x) \rightarrow \text{mortal}(x))$

$\forall x(\text{Dog}(x) \rightarrow \text{mortal}(x))$

Person(小张)

Person(小李)

这些事实表示,所有的人都必然会死,所有的狗也必然会死,小张和小李是人。如果给出的推理规则为肯定前件的假言推理,则人们很容易推出小张最终必然会死。

采用陈述性表示的**知识库**具有如下特征:其中的知识是一个个独立的知识片,它们在推理机的控制下通过数据基的相互作用来达到问题的求解。知识库的更改是通过增加、删除和修改知识片实现的。

在知识表示方法中,向来就有陈述性表示和**过程性表示**两大类。陈述性表示只给出事物本身的属性及事物之间的相互关系,对问题的解答就隐含在这些知识之中。而过程性表示则给出解决一个问题的具体过程。两者相比,陈述性表示比较简要,清晰,可靠,便于修改,但往往效率低。过程性表示则与此相反,它比较直截了当,效率高,但由于详细地给出了解题过程,使这种知识表示显得复杂,不直观,容易出错,不便修改。

但是,在争论究竟是过程性表示更好还是陈述性表示更好时,我们不能忘了一个事实,即陈述性表示和过程性表示实际上并没有绝对分界线。任何陈述性表示如果要被实际使用,必须有一个相应的过程去解释执行它。对于一个以使用陈述性表示为主的系统来说,这种过程往往隐含在系统之中而不是

面向用户。用户所看见的只是陈述性表示，所以，陈述性表示和过程性表示的区别实际上在于：第一，解题的具体过程和算法是直接面向用户（用户能看见这种算法，能选用算法，修改算法，甚至自己设计算法），还是作为系统固有的设备而隐藏在系统之中（使用户必须接受先天决定的解题过程）。第二，解题过程是通用的（可应用于某种特定的陈述性语言表示的任何知识），还是专用的（只能解决某个特定问题）。

参考文献

1. 陆汝钤. 人工智能（上）. 北京：科学出版社，1989

2. 管纪文，刘大有，等. 知识工程原理. 长春：吉林大学出版社，1988

（刘大有　唐海鹰　虞强源）

chengzai wangluo

承载网络（carrier network）　作为通信网络的基础设施，传统电信服务提供商通过规划、建设和运营承载网络，为语音和数据的通信提供长途、大容量、高可靠性的传输保障。承载网络通过端口局进行两两互连或集中到交换中心互联，实现跨运营商的信息互通。依据通信网络的覆盖范围，传统电信服务提供商一般利用光纤和微波通信构成跨洲际，或全国性、地区性的大容量骨干传输网。近年来，光纤通信已经成为骨干传输网的主要通信支撑技术。

骨干传输网承载的业务类型在过去主要是以语音通话为主。为社会公众和机构，以及中小型专业通信公司提供遍及跨越广阔地域空间的、相互关联互通的、大容量和高可靠的有线/移动电话等传统电信、广播电视和互联网服务以及企业、政府等专业网络的服务。

随着技术的进步，语音通信已经全面实现数字化传输，语音通信和数据通信将通过复用骨干传输网络的信道资源实现信息的传输。随着基于分组交换的 TCP/IP 技术在全球范围内得到快速发展，与语音通信所占传输容量相比较，互联网使用的传输容量在骨干传输网中的比重越来越大。

参考文献

吴德本. 新编电信技术概论. 北京：人民邮电出版社，2003

（马严）

chengfaqi

乘法器（multiplier）　对以数字形式表示的两个 n 位数求积的一种运算电路。

两个原码数相乘，其乘积的数值为两数绝对值之积，而符号为两数符号的异或值。设 x、y 为被乘数与乘数，x_S、y_S 为被乘数与乘数的符号，则其乘积 $p=|x|\times|y|$，符号 $p_s=x_s\oplus y_s$。

设 $x=0.1101$，$y=0.1011$，人工计算乘积 $x\cdot y$ 的过程如下

$$
\begin{array}{r}
0.\;1\;1\;0\;1\\
\times\,0.\;1\;0\;1\;1\\
\hline
1\;1\;0\;1\\
1\;1\;0\;1\;\;\\
0\;0\;0\;0\;\;\;\;\\
1\;1\;0\;1\;\;\;\;\;\;\\
\hline
0.1\;0\;0\;0\;1\;1\;1\;1
\end{array}
$$

即 $x\cdot y=0.10001111$，符号为正。

计算机内实现原码一位乘法的逻辑电路如图 1 所示，其中三个寄存器 A，B，C 分别存放部分积、被乘数和乘数，计数器用来控制累加与移位次数。原码一位乘法流程如图 2 所示，其中 CR 为计数器，C_0 为 C 寄存器的最低位。运算方法如下：

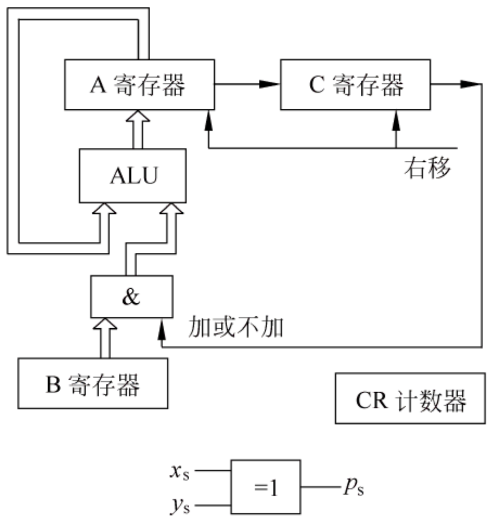


图 1　原码一位乘法逻辑电路

（1）在计算机内多个数据一般不能同时相加，一次加法操作只能求出两数之和，因此每求得一个相加数，就与上次部分积相加。

（2）人工计算时，相加数逐次偏移 1 位，而最后的乘积位数是乘数（或被乘数）位数的 2 倍，如按这种方法在计算机中运算，加法器也要加长 1 倍。从计算机的计算过程可以发现，在求本次部分积时，前一次部分积的最低位，不再参与运算，因此可将其右

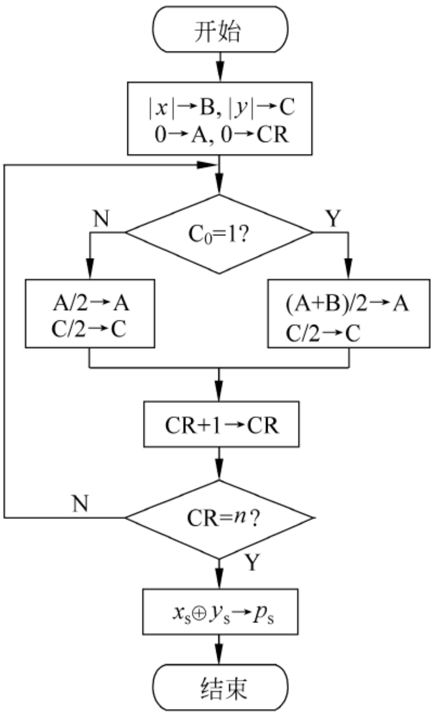


图 2　原码一位乘法流程

移一位,相加数可直送而不必偏移,于是用 n 位加法器就可实现两个 n 位数相乘。

(3) 部分积右移时,乘数寄存器 C 同时右移一位,这样可以用乘数寄存器的最低位来控制相加数(取被乘数或零),同时乘数寄存器 C 的最高位可接收部分积右移出来的一位,因此,完成乘法运算后,A 寄存器中保存乘积的高位部分,C 寄存器中保存乘积的低位部分。

上述的乘法运算是采用串行移位和并行相加的方法,这种方法不需要很多的硬件,然而累加-移位的方法毕竟太慢,对于 $n \times n$ 位乘法,就需要 n 次累加和移位,即使是采用每次乘两位的乘法,将乘法速度提高了一倍,仍然嫌慢。

自从大规模集成电路问世以来,高速的单元阵列乘法器应运而生,为高速乘法的实现提供了硬件基础。另一方面,优化乘法算法,压缩累加的次数,也是提高乘法速度的好方法。图 3 示出了一个阵列乘法器,可完成 $x \cdot y$ 乘法运算($x = x_1x_2x_3x_4$, $y = y_1y_2y_3y_4$)。阵列的每一行送入乘数 y 的每一数位,每一列则送入被乘数 x 的每一数位。图中每一个方框包括一个与门和一位全加器。该方案所用加法器数量较多,但内部结构规则性强,适于用大规模集成电路实现。

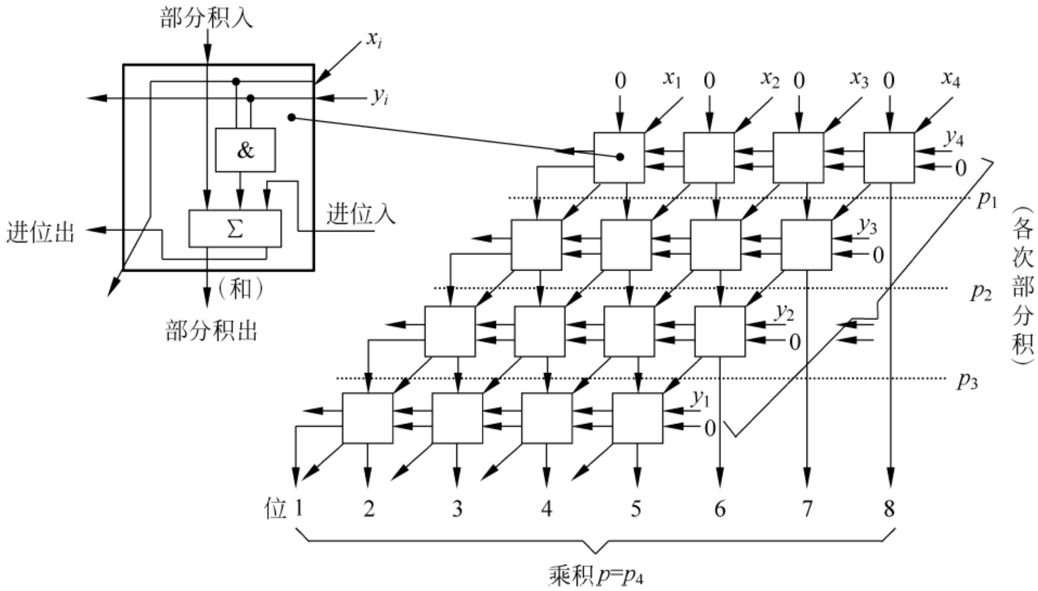


图 3　阵列乘法器

参考文献

1. 蒋本珊. 电子计算机组成原理. 北京:北京理工大学出版社,1994

2. 王爱英. 计算机组成与结构. 3 版. 北京:清华大学出版社,2001

(刘恩德)

chengxu

程序(program)　计算任务的处理对象和处理规则的描述。任何以计算机为处理工具的任务都是计算任务。处理对象是数据(如数字、文字、图形、图像、声音等,它们只是表示,而无含义)或信息(赋

予数据的含义)。处理规则一般指处理动作和步骤。在**低级语言**中,程序是一组指令和有关的数据或信息。在高级语言中,程序一般是一组**说明**和**语句**。程序是程序设计中最基本的概念,也是软件中最基本的概念。程序是软件的本体,又是软件的研究对象。程序的质量决定软件的质量。以上是在实现级语言中程序的含义。在设计级语言中,程序即设计规约,在功能级语言中,程序即功能规约,在需求级语言中,程序即需求定义。但习惯上,程序均指实现级语言中的程序。

程序要能实际起作用,必须装入到机器内部。程序的实际工作过程称为程序的执行。衡量程序质量,除对程序结构进行静态考察外,还必须考察其执行过程。与执行过程无关的特性称为程序的静态特性;与执行过程有关的特性称为程序的动态特性。

发展过程

在软件发展的第一阶段(1946—1956 年),程序都是用**机器语言**或接近于机器语言的**汇编语言**书写的,即都是低级语言程序,从内部特性上看,程序内部的工作严格依顺序执行,因此,都是顺序程序。衡量程序质量的标准主要是功效,运行时间要省,占用空间要小。在软件发展的第二阶段(1956—1968 年),程序主要都用**高级语言**书写,即高级语言程序。当然,低级语言程序仍然存在。这时除了顺序程序以外,还出现了具有并行成分的并发程序和并行程序。衡量程序质量的标准,已经逐步转向易读性和易维护性。在软件发展的第三阶段(1968 年以来),由于程序的规模增大,对程序的模块化、结构化的要求越来越高,出现了一些模块化语言。同时,由于并发程序的比重增高,为了更好地书写并发程序,出现了一些具有并行成分的语言,并且由于实时处理的需要,在语言中设有相应的实时处理成分。总之,这一阶段的程序,主要是具有并行成分和实时处理成分的模块化程序,即现代高级语言程序。衡量程序质量的标准主要是结构良好性,使之易读、易维护。

基本成分

构成程序的基本成分包括**子程序**、子例程、**例程**、**协同例程**、递归例程、模块和构件,它们均称为程序单位。

子程序　与子计算任务相应的处理对象和处理规则的描述。

子例程　可由其他程序或子程序调用的子程序。子例程有两个方面:一个是定义方面,称为子例程定义或子例程说明;另一个是调用方面,称为子例程调用。随着实现方式的不同,又可区分为开式子例程和闭式子例程。二者各有利弊。开式子例程时间节省,空间浪费;闭式子例程恰恰相反。

例程　子例程的同义语。

协同例程　一组可以互相调用的程序单位,它们彼此处于平等地位,调用后无须返回到开始位置,且自带工作区。

递归例程　可以作为其本身的子例程而被调用的例程。这种调用可以是直接的,也可以是间接的(即通过其他子例程)。

模块　具有相对独立性的一组逻辑上有关的实体。在现代**高级语言**中,有各种定义模块的方式,但其主要成分是一组声明和一组**语句**。

构件　具有封装性、复用性和组装性的程序单位。

参考文献

1. 徐家福. 系统程序设计语言. 北京:科学出版社,1983

2. Sammet J E. Programming languages: history and fundamentals. Englewood Cliffs, NJ: Prentice Hall,1969

(徐家福)

chengxu jishuqi

程序计数器(program counter)　**程序**运行中指明下一条待执行指令所在地址的一种带有计数功能的指令地址寄存器,又称**指令计数器**。当一条现行指令执行完毕的时候,程序计数器作为指令地址寄存器,其内容必须已经改变成下一条指令的地址,从而使程序得以持续运行。为此可采取以下两种办法。

第一种办法是在指令中除了规定操作数的地址外也规定了下一条指令的地址。在现行指令执行过程中将这个地址送入指令地址寄存器即可达到目的。早期以磁鼓、延迟线等串行装置作为主存储器的计算机多采用此法,因为根据本条指令的执行时间恰当地选定下一条指令的地址可以缩短读取指令的串行等待时间,从而收到提高程序运行速度的效果。

第二种办法是顺序执行指令。一个程序由若干个程序段组成,每个程序段的指令可以设计成顺序地存放在存储器之中,所以只要指令地址寄存器兼有计数功能,在执行指令的过程中进行计数,自动加

一个增量,就可以形成下一条指令的地址,从而达到顺序执行指令的目的。这个办法适用于以随机存取装置作为主存储器的计算机。当程序的运行需要从一个程序段转向另一个程序段时,可以利用转移指令来实现。转移指令中包含了即将转去的程序段入口指令的地址。执行转移指令时将这个地址送入程序计数器(此时只作为指令地址寄存器,不计数)作为下一条指令的地址,从而达到转移程序段的目的。子程序的调用、中断和陷阱的处理等都用类似的方法。

在随机存取存储器普及以后,第二种办法的整体运行效果大大地优于第一种办法,因而顺序执行指令已经成为当代主流计算机普遍采用的办法,程序计数器就成为**中央处理器**不可或缺的一个控制部件。 (张梓昌)

chengxu lilun

程序理论(theory of programs) 研究**程序**的语义性质和程序的设计及开发方法的理论。主要包括程序语义理论、数据类型理论、程序逻辑理论、程序验证理论、并发程序设计理论和混合程序设计理论。程序理论和算法理论是计算机科学的两大支柱。

程序理论的基本问题是如何建立一个相对完善的理论框架,为软件的设计和开发方法提供理论依据。这个框架应能提供有效地描述程序规约的语言;应能定义可操作的变换方法以便能规约构造可执行的程序;应能给出验证程序与其规约之间一致性的机制。

程序是用程序语言编写的,研究程序的规约、变换和验证,必须首先给出程序语言的语义。这种语义用数学方法刻画程序语句的加工过程,并将其执行结果形式化。所以,程序语义也叫**形式语义**。形式语义的研究始于20世纪60年代初期,ALGOL 60是第一个明确区分语法和语义的程序设计语言。J. McCarthy, P. J. Landin, C. Strachy, D. Scott, C. A. R. Hoare和E. W. Dijkstra等学者,以及爱丁堡大学和维也纳IBM研究中心的计算机科学家们对程序语义的研究和发展都起过重要作用。

程序语言的形式语义分为四类:①**操作语义**,模拟程序执行中计算系统的操作过程。②**指称语义**,把程序作为论域间的泛函以便刻画程序的执行数学结果。③**公理语义**,用公理化方法刻画程序与被加工数据的逻辑关系。④**代数语义**,把程序执行的结果定义为满足某种公理体系的代数结构。程序理论对形式语义的需求,促进了论域、偏序以及范畴论等数学理论的发展;而形式语义理论的研究又促进了程序语言和程序设计方法的进步,例如:在高级语言中广泛使用的过程说明和过程调用的精确概念和实现机制,就是在语义理论的研究中逐步明确的。

程序的设计和开发理论早期研究的目标是解决程序验证问题。C. A. R. Hoare在20世纪60年代首先提出了**程序逻辑**的理论。这个理论的基本逻辑公式形式为

$$\{\phi\}S\{\psi\}$$

其中,ϕ,ψ 是关于程序变元的逻辑表达式,ϕ 称为前置条件,ψ 称为后置条件。公式 $\{\phi\}S\{\psi\}$ 为真当且仅当:如果程序 S 执行前程序变元满足 ϕ,则程序执行后程序变元满足 ψ。程序设计的目标就是构造使 $\{\phi\}S\{\psi\}$ 成立的程序。C. A. R. Hoare对程序语言的每个语句都给出了相应的逻辑规则。因此,在程序 S 给定后,可以使用这些规则证明:如果程序 S 执行前程序变元满足 ϕ,则程序执行后程序变元满足 ψ。

霍尔逻辑的缺点是其基本形式不能进行逻辑演算。为了克服这种缺点,人们在20世纪80年代,提出了类型论方法以求为程序的设计和开发建立更完善的理论框架(见**类型理论**)。比较典型的类型理论是由P. Martin-Löf建立的,称为直觉主义类型论(见**马丁洛夫类型理论**)。它的基本思想是:精选一组类型(集合)作为程序规约,而它们的元素就是满足规约的程序;用一组规则定义类型与其元素间的隶属关系,这些规则即是从规约产生程序的变换规则,又是一阶(直觉主义)逻辑的证明规则。因此,只要对给定的规约(逻辑命题)进行证明,就可以构造出符合此规约的程序。这样,程序规约、变换、验证都寓于对规约的证明之中了。马丁洛夫理论的基本对象是 $a: A$,其中 A 是一类型,a 是类型 A 中的元素,称为 A 的居元。这个理论定义了几种基本类型:$A \to B, A \times B, A + B, \Pi_{x:A} B$ 和 $\Sigma_{x:A} B$,每种类型都由一组规则定义,这些规则确定了居元和类型间的关系。这个理论的基本对象 $a: A$ 有多种解释,这些解释导致了它在程序设计和开发理论中的应用。例如:$a: A$ 可以解释为(居元:类型)、(证明:命题)、(程序:规约)等。以类型 $A \to B$ 为例,它有下述两条基本规则:

$$\frac{x: A \vdash b: B}{\lambda x.\, b: A \to B} \qquad (1)$$

$$\frac{m: A \to B \quad n: A}{mn: B} \tag{2}$$

它们有三种解释：

解释1：（居元：类型）。规则(1)表示：若类型 A 有居元 x 可以推出类型 B 有居元 b，则类型 $A \to B$ 有居元 $\lambda x.b$。规则(2)表示：若类型 $A \to B$ 有居元 m 且类型 A 有居元 n，则类型 B 有居元 mn。其中，$\lambda x.b$，mn，m 和 n 称为 λ-项，前两者分别称为 λ-抽象和 λ-作用（见 **λ 演算**）。类型理论的重要结论是：给定典型居元 a 和类型 A，可以在有限步内判断 a 是否为 A 的居元，这就是类型理论的强范式化性质。

解释2：（证明：命题）。采用直觉主义逻辑的观点，一个逻辑命题为真当且仅当存在此命题的证明。若把 $a: A$ 解释为 a 是命题 A 的证明，把类型 $A \to B$ 解释为 A 蕴涵 B。则规则(1)表示：若命题 A 有证明 x 可以推出命题 B 有证明 b，则命题 $A \to B$ 有证明 $\lambda x.b$。规则(2)表示：若 $A \to B$ 有证明 m 且命题 A 有证明 n，则命题 B 有证明 mn。若把规则(1)和(2)中的所有“证明”（类型的居元）去掉，它们就成为关于逻辑蕴涵的自然推理规则：

$$\frac{A \vdash B}{A \to B} \tag{1'}$$

$$\frac{A \to B \quad A}{B} \tag{2'}$$

类似地，若把类型 $A \times B$，$A + B$，$\Pi_{x:A}B$ 和 $\Sigma_{x:A}B$ 分别解释为 $A \wedge B$，$A \vee B$，$\forall x.B$ 和 $\exists x.B$，则关于这些类型的规则就成为关于 $\wedge$（与），$\vee$（或），$\forall$（全称量词）和 $\exists$（存在量词）的逻辑推理规则。这就是 Howard 的“命题即类型”原则。

解释3：（程序：规约）。假定一阶逻辑语言作为规约语言，函数式语言作为程序语言，则 λ-项：$\lambda x.b$，mn，m 和 n 都将表示程序，而 $a: A$ 可以解释为程序 a 满足规约 A。规则(1)和(2)在这种情况下定义了程序与规约间的关系。规则(1)表示：若程序 x 满足规约 A 可以推出程序 b 满足规约 B，则程序 $\lambda x.b$ 满足规约 $A \to B$。规则(2)表示：若程序 m 满足规约 $A \to B$，且程序 n 满足规约 A，则程序 mn 满足规约 B。

根据这三种解释，对程序理论的基本问题：给定规约 A，要找出满足它的程序 a，就可做下述新的理解：①按照解释3，规约 A 是一逻辑表达式。对它可以使用逻辑推理规则进行证明。②证明的过程就是引用逻辑推理规则的过程。③又根据解释2和3，逻辑推理规则就是程序构造规则。在证明中，引用一条规则的同时，就构造出满足被证明的某个子公式的一个程序片段。④如果规约 A 被证明为真，在证明过程中就构造出满足规约 A 的一个程序。总之，根据马丁洛夫理论，只要将作为规约的逻辑公式证明为真，就可构造出满足规约 A 的程序。因此，程序验证作为程序理论的单独组成部分就不再需要了。这就是将类型理论作为程序设计和开发理论的典型方法。

根据这种思想，人们在20世纪80年代推出了多种不同的交互式证明编辑工具，使用这类工具时，对规约的证明是由用户完成的。用户引用开发工具中的逻辑推理规则，对规约进行证明，并构造程序。这种引用的合理性是由计算机检查的，类型理论的强范式化性质保证了这种检查的可行性。目前，这类开发工具尽管保证了所开发程序的正确性，但它们的程序设计质量与人工设计的程序质量相比差距很大，尚待提高。

20世纪70年代初期，为了保证在操作系统中多个并行执行进程的正确性，导致了并发程序理论的产生。进入80年代以来，随着超大规模集成电路技术的日臻成熟，并行和分布计算机系统得到了迅速发展，特别是互联网的出现和广泛使用，大大促进了并行程序理论的发展，使之成为程序理论的一个重要分支。

并发程序是包含多个没有因果关系的进程的程序。进程间的通信、同步和它们的并行执行是并发程序区别于顺序程序的基本操作。计算机科学中的并发概念是由 Petri 在 1962 年提出的。Hoare 和 Milner 在20世纪70年代后期，分别提出并发程序的 CCS 和 CSP 模型，他们把输入输出以及并行执行作为基本语法成分引入程序设计语言，并使用结构操作语义方法定义模型中基本语法成分的操作语义，开创了用代数方法研究并发程序中通信和并行行为的领域。并发程序理论研究的内容包括：如何刻画并行进程的行为，在什么情况下它们可以互相模拟，研究各种通信及同步机制，以及死锁及活性，可观察性和发散性等并发现象。并发程序理论的研究加深了人们对并发系统的认识；其主要研究成果，例如，关于通信和同步的概念，已作为基本语言成分在 Ada，Occam 和 Java 等程序语言中得到广泛应用。

进入20世纪90年代，对控制系统的研究，例如，对自动导航及核电站监测等控制系统的研究，引起了程序理论界的关注。这些系统的特点是：它们

都使用计算机进行实时控制；对控制系统的安全性和可靠性要求高，控制系统的微小错误都将给经济和社会安全带来巨大的损失；系统中都存在两类不同对象：根据传统控制理论，使用微分方程刻画的连续现象和根据逻辑或代数方法，使用计算机程序控制的离散型事件。这类系统又叫混合系统。近年来，混合系统的程序理论已成为研究热点。其基本目标是：建立混合系统的计算模型，设计描述混合系统的高级语言，探索混合系统程序的设计和开发方法。目前，在建立混合计算模型方面有三种不同的方法：第一种称为逻辑型混合计算模型方法，其基本思想是在时态逻辑基础上引入时段和切变的概念。时段用于刻画系统在时间区间上的连续变化，切变则表示系统中离散事件间的时序关系。第二种是程序设计型混合模型方法，其目标是在 CSP 及 ADA 等并发语言中引入连续变量及给定初值的微分方程，而语言中原有的通信、顺序及条件语句则用来表示系统中离散事件的关系。第三种方法是将自动机概念推广，把微分方程刻画的连续现象扩充为自动机的状态，用自动机状态转移刻画离散事件间的关系。尽管混合系统理论发展的历史不长，但它在一些重要的实时控制系统中已经得到了应用，显示出这些理论的生命力。

参考文献

1. ACM Turing Award Lectures: The first twenty years, 1966 to 1985. Reading, MA: Addison-Wesley, 1987

2. Martin-Löf P. Intrutionistic type theory. Bibliopolis, Napels, 1984

（李未）

chengxu luoji

程序逻辑（program logic） 描述和论证程序行为的逻辑。

由于程序和逻辑有着本质的联系，因此程序逻辑的研究一直是计算机科学最活跃的领域之一。美国 R. W. Floyd 于 20 世纪 60 年代中期把程序框图与逻辑公式相对应，提出使用逻辑描述和分析程序的想法。20 世纪 60 年代末期，英国 C. A. R. Hoare 首次给出一个程序语言的逻辑系统，提出程序部分正确性的形式验证规则（见**程序验证**）。20 世纪 70 年代一些科学家提出用**模态逻辑**和**时态逻辑**描述和论证程序。这些工作推动了程序逻辑的研究。

程序逻辑起源于验证程序正确性的需求，用逻辑公式描述对输入和输出信息的要求，建立逻辑公式与程序间的联系，表示为：

$$\{P\}S\{Q\}$$

其中 P 和 Q 为有关程序变元的逻辑表达式；P 称为 S 的前置条件，Q 称为 S 的后置条件。此公式表示，如果程序 S 执行前程序变量的值满足前置条件 P，且程序终止，则程序 S 执行完成时，程序变量的值满足后置条件 Q。可以建立一套关于这类公式的推理规则，得到一个描述程序行为的逻辑系统，这就是著名的霍尔逻辑。这种逻辑的缺点是逻辑公式不能描述程序 S 的终止特性，因此它是讨论程序部分正确性的逻辑。如果在霍尔逻辑系统中又能证明对满足前置条件的所有输入变量，程序 S 都终止，则程序具有完全正确性。

在公式 $\{P\}S\{Q\}$ 中，逻辑表达式 P 和 Q 实际上是当作语句 S 的注释使用，对逻辑公式 $\{P\}S\{Q\}$ 不能直接进行逻辑运算，如 $\{P_1\}S\{Q_1\}\rightarrow\{P_2\}S_2\{Q_2\}$ 就没有意义，因此霍尔逻辑还没有把程序和逻辑统一起来。

解决的方法之一就是在逻辑系统中引入模态词来刻画程序的动态行为，用模态逻辑描述程序行为。

例如，引入模态词[ ]，而 $[S]Q$ 表示当 S 终止时 Q 为真；引入模态词⟨ ⟩，而 $\langle S\rangle Q$ 表示存在一个时刻，S 终止且 Q 为真。这样程序 S 的部分正确性就可以表示为 $P\rightarrow[S]Q$，即 P 蕴涵当 S 终止时 Q 为真；而 $P\rightarrow\langle S\rangle Q$，则表示 S 的完全正确性。

20 世纪 70 年代中期开始，并发式程序设计逐渐成为程序理论的主要研究课题之一。而不同任务间的同步和信息交换及有关死锁等是并发程序不同于顺序程序的主要的动态特性。

在逻辑系统中，引入时态联结词描述程序的动态特性是一种自然的解决方法，例如可以引入◇，$\Diamond P$ 表示在将来某一时刻 P 真；引入联结词□，$\Box P$ 表示 P 从此以后真。使用时态逻辑可以对每个程序构造出它所对应的逻辑系统，并可以在这个系统中刻画程序终止（记作 STOP）及无死锁等概念。例如程序 S 的部分正确性问题就可以表示为 $P\rightarrow\Box(\mathrm{STOP}\rightarrow Q)$。它表示程序开始执行时，$P$ 真蕴含下述论断：如果将来某一时刻程序终止则 Q 真。程序的完全正确性则可表示为 $P\rightarrow\Diamond(\mathrm{STOP}\wedge Q)$，它表示程序开始执行时 P 真蕴含下述命题：存在某一时刻 S 终止并且 Q 真。

软件工程的一个重要方法是在软件开发前首先把程序要达到的目标即规约描述清楚，然后建立一套系统地从功能描述到执行程序的开发和检验技

术。功能描述应当简洁易懂,开发方法应当有效,易于验证。程序逻辑已成为联结功能规约和开发方法及进行程序验证的一个重要手段。

参考文献

1. 陆汝钤. 计算系统的形式语义. 北京: 清华大学出版社,2017

2. Manna Z. Mathematical theory of computation. McGraw-Hill,1974

(李未)

chengxu sheji

程序设计(programming)　设计、编制和调试**程序**的方法与过程,或研究、开发上述方法与过程中所涉及的理论、原则、方法及技术所涉及的学科。又称编程。这里的程序一般是指实现级语言的程序。它是目标明确的智力活动。由于程序是软件的本体,软件的质量主要是通过程序的质量来体现的,程序设计工作在软件研究中的地位就显得非常重要,内容涉及有关的基本概念、工具、方法以及方法学等。

分　类

按照结构性质,有**结构化程序设计**与非结构化程序设计之分。前者指的是具有结构性的程序设计方法与过程。它具有由基本结构构作复杂结构的层次性,后者反之。按照用户要求,有**过程式程序设计**与非过程式程序设计之分。前者指的是使用过程式程序设计语言的程序设计;后者则指使用非过程式程序设计语言的程序设计;按照程序的成分性质,有**顺序程序设计**、**并发程序设计**、**并行程序设计**、**分布式程序设计**之分。顺序程序设计是设计、编制和调试顺序程序的方法与过程。并发程序设计是设计、编制和调试并发程序的方法与过程;并行程序设计、分布式程序设计的含义依此类推。按照设计风格,有**逻辑式程序设计**、**函数式程序设计**、对象式程序设计(**面向对象程序设计**)之分。逻辑式程序设计是以逻辑子句为基本构件的程序设计;函数式程序设计是以函数为基本构件的程序设计;对象式程序设计是以对象类为基本构件的程序设计。此外,尚有**可视程序设计**、**文化程序设计**等。

基本内容

程序设计的基本概念有程序、数据、子程序、子例程、协同例程、模块和构件以及顺序性、并发性、并行性和分布性等。程序是程序设计中最为基本的概念。子程序、子例程和协同例程等都是为了便于进行程序设计而建立的程序基本单位。顺序性、并发性、并行性和分布性反映程序的内在特性。

程序设计规范是进行程序设计的具体规定。程序设计是软件开发工作的重要部分,而软件开发是工程性的工作,所以要有规范。程序设计工具包括用以书写程序的语言和为了便于进行程序设计而提供的各种专用程序等。语言影响程序设计的功效,以及软件的可靠性、易读性和易维护性。专用程序为软件人员提供合适的环境,便于进行程序设计工作。

程序设计方法有两类。一类是全局性的,如结构化程序设计方法。它不仅要求编出的程序结构良好,而且要求程序设计过程是结构化的、层次式的、逐层降低抽象级别的。另一类则是局部性的。如子例程方法、协同例程方法等。全局性的方法与规范的关系密切,而且互有影响。

程序设计的发展可归结为从顺序程序设计到并发程序设计、并行程序设计、分布程序设计;从非结构化程序设计到结构化程序设计;从过程式程序设计到非过程式程序设计,到逻辑式程序设计、函数式程序设计、对象式程序设计,以及可视程序设计、文化程序设计等;从低级语言工具到高级语言工具。

参考文献

1. 徐家福. 系统程序设计语言. 北京: 科学出版社,1983

2. 徐家福. 对象式程序设计语言. 南京: 南京大学出版社,1992

3. Sammet J E. Programming languages: history and fundamentals. Englewood Cliffs, NJ: Prentice Hall,1969

(徐家福)

chengxu sheji fangfaxue

程序设计方法学(programming methodology)　以程序设计方法为研究对象的学科。它主要涉及用于指导程序设计工作的原理和原则,以及基于这些原理和原则的设计方法和技术,着重研究各种方法的共性与个性,各自的优缺点。一方面,要涉及方法的理论基础与形成背景;另一方面,也要涉及方法的基本架构与实用价值。程序设计方法学的另一种含义是,针对某一领域或某一领域的特定一类问题进行程序设计的指导原则和所用的一整套特定程序设计方法所构成的体系。例如,基于 Ada 程序设计语言的程序设计方法学。

作为一门学科(第一种含义),程序设计方法学

目前系统性的研究成果尚不多,但由于它可对程序设计人员选用具体的程序设计方法起指导作用,而具体的程序设计方法对程序设计工作的质量以及所设计出的程序的质量影响巨大,因此,程序设计方法学研究的重要性也就不言而喻。

作为进行程序设计的指导原则和一整套特定程序设计方法所构成的体系(第二种含义),目前已出现多种程序设计方法学。例如,各种逻辑式程序设计方法学、函数式程序设计方法学、对象式程序设计方法学等。它们各自的利弊得失均和具体领域、具体问题以及具体环境有关。上述两种含义的关系是,第二种含义是第一种含义的基础,第一种含义是在第二种含义的基础上的总结、提高,上升到原则、原理和理论的高度。因此,这两种含义的程序设计方法学都很重要。它们既对实际程序设计工作有指导意义,又对软件的发展有较大影响。 (徐家福)

chengxu sheji yuyan

程序设计语言(programming language) 用于书写计算机程序(习惯上指实现级语言程序)的语言。语言的基础是一组记号和一组规则。根据规则由记号构成的记号串的总体就是语言。在程序设计语言中,这些记号串就是**程序**。程序设计语言包含三个方面,即**语法**、**语义**和**语用**。语法表示程序的结构或形式,亦即表示构成语言的各个记号之间的组合规则,但不涉及这些记号固有的以及和使用情景有关的含义。语义表示程序的固有含义,亦即表示按照各种方法所表示各个记号的特定含义,但不涉及使用者。语用表示程序与使用情景有关的含义。

语言的好坏不仅影响到程序人员使用是否方便,而且涉及程序人员所写程序的质量。

基本成分 语言的种类千差万别。但是,一般说来,基本成分不外四种。①数据成分,用以描述程序中所涉及的数据;②运算成分,用以描述程序中所包含的运算;③控制成分,用以表达程序中的控制构造;④传输成分,用以表达程序中数据的传输。

分类 按照语言级别,有**低级语言**与**高级语言**之分。低级语言包括字位码、**机器语言**和**汇编语言**。它的特点是与特定的机器有关,功效高,但使用复杂、烦琐、费时、易出差错。其中字位码是计算机唯一可直接理解的语言,但由于它是一连串的字位,复杂、烦琐、冗长,几乎无人直接使用。机器语言是表示成数码形式的机器基本指令集,或者是操作码经过符号化的基本指令集。汇编语言是机器语言中地址部分符号化的结果,或进一步包括宏构造。高级语言的表示方法要比低级语言更接近于待解问题的表示方法,其特点是在一定程度上与具体机器无关,易学、易用、易维护。当高级语言程序翻译成相应的低级语言程序时,一般说来,一个高级语言程序单位要对应多条机器指令,相应的编译程序所产生的目标程序往往功效较低。

按照用户要求,有**过程式语言**和**非过程式语言**之分。过程式语言的主要特征是,用户可以指明一列可顺序执行的运算,以表示相应的计算过程。例如,FORTRAN,COBOL,ALGOL 60 等都是过程式语言。非过程式语言的含义是相对的,凡是用户无法指明表示计算过程的一列可顺序执行的运算的语言,都是非过程式语言。著名的例子是 PROLOG 和 RPG。

按照应用范围,有通用语言和专用语言之分。目标非单一的语言称为通用语言,例如,FORTRAN,COBOL,ALGOL 60 等都是通用语言。目标单一的语言称为专用语言,如 APT 等。

按照使用方式,有**交互式语言**和非交互式语言之分。具有反映人机交互作用的语言成分的语言称为交互式语言,如 BASIC 语言就是交互式语言。语言成分不反映人机交互作用的语言称为非交互式语言,如 FORTRAN,COBOL,ALGOL 60,PASCAL,C 等都是非交互式语言。

按照成分性质,有顺序语言、并发语言、并行语言和分布语言之分。只含顺序成分的语言称为顺序语言,如 FORTRAN,PASCAL,C 等都是顺序语言。含有并发成分的语言称为并发语言,如并发 PASCAL,Modula 和 Ada 等都是并发语言。含有并行成分的语言称为并行语言。考虑到分布计算要求的语言称为分布语言,如 Modula * 便是分布语言。

传统的程序设计语言大都以冯·诺依曼式的计算机为设计背景,因而又称为冯·诺依曼式语言。J. Backus 于 1977 年提出的函数式语言 FP 则以非冯·诺依曼式的计算机为设计背景,因而又称为非冯·诺依曼式语言。

主要语言举例

(1) APT——自动数控程序语言 第一个专用语言,用于数控机床加工,1956 年。

(2) FORTRAN——公式翻译程序设计语言 第一个广泛使用的高级语言,为广大科学和工程技术人员使用计算机创造了条件,1956 年。

(3) FLOW-MATIC 第一个适用于商用数据处理的语言,其语法与英语语法类似,1956 年。

(4) IPL-V——信息处理语言-V　第一个表处理语言,它可看成是一种适用于表处理的假想计算机上的汇编语言,1958 年。

(5) COMIT——麻省理工学院编译程序语言　第一个串处理和模式匹配语言,1957 年。

(6) COBOL——面向商业的通用语言　使用最广泛的商用语言,1960 年。

(7) ALGOL 60——算法语言 60　程序设计语言由技艺转向科学的重要标志,其特点是局部性、动态性、递归性和严谨性,1960 年。

(8) LISP——表处理语言　引进函数式程序设计概念和表处理设施,在人工智能领域内广泛使用,1960 年。

(9) JOVIAL——国际代数语言的朱尔斯文本　第一个具有处理科学计算、输入输出逻辑信息、数据存储和处理等综合功能的语言。多数 JOVIAL 编译程序都是用 JOVIAL 书写的,1960 年。

(10) GPSS——通用系统模拟语言　第一个使模拟成为实用工具的语言,1961 年。

(11) APL　一种提供很多高级运算符的语言,它可使程序人员写出甚为紧凑的程序,特别是涉及矩阵计算的程序,但其实现文本到 1967 年才有定义,1962 年。

(12) JOSS——Johnniac 开放系统　第一个交互式语言,它有很多方言,曾使分时成为实用,1964 年。

(13) FORMAC——公式处理编译程序　第一个广泛用于需要形式代数处理的数学问题领域的语言,1964 年。

(14) SIMULA——模拟语言　一种主要用于模拟的语言,它是 ALGOL 60 的扩充,1966 年。SIMULA 67是 1967 年 SIMULA 的改进,其中引进了"对象"及"类"等概念,它是第一个对象式语言。

(15) PASCAL——菲利浦自动顺序计算机语言　它是在 ALGOL 60 的基础上发展起来的重要语言,其最大特点是简明性与结构性,1971 年。

(16) PROLOG　一种处理逻辑问题的语言。它已广泛用于**关系数据库**、数理逻辑、抽象问题求解、自然语言理解等多个领域,1971 年。

(17) Smalltalk　一种面向对象程序设计语言。自从 1971 年出现后,曾有多种不同文本,其中使用最广的是 Smalltalk 80,其显著特点是利用"对象",以使面向对象程序设计得到广泛应用,1971 年。

(18) C　一种使用颇为广泛的程序设计语言。它原先是为辅助开发 UNIX 操作系统而设计的,后来广泛用于研究、开发与教学,主要用于系统程序设计,同时也用于其他领域,1973 年。

(19) Ada　一种现代模块化语言,它属于 ALGOL-PASCAL语言族,但有较大变动。其主要特征是强类型化和模块化,便于实现分别编译,提供类属设施与异常处理,适于嵌入式应用,1979 年。

除了上面列举的语言外,还有一些较为通用的语言,特别是 BASIC,PL/1,SNOBOL,ALGOL 68 等。BASIC 虽然简单、易学,使用广泛,但其中没有什么新概念,而且并不是第一个交互式语言。PL/1 的设计思想来源于 JOVIAL,其功能来源于 FORTRAN,COBOL,ALGOL 60,具有中断和表处理等设施。SNOBOL 是一种好的语言,对 COMIT 中若干概念做了明显的改进。ALGOL 68 在语言成分和描述方法方面虽有所创新,但应用尚不广泛。

发展趋势　程序设计语言是软件的重要方面。它的发展趋势是模块化、简明性、形式化、并行化和可视化。①模块化,不仅语言具有模块成分,程序由模块组成,而且语言本身的结构也是模块化的。②简明性,涉及的基本概念不多,成分简单,结构清晰,易学易用。③形式化,发展合适的形式体系,以描述语言的**语法**、**语义**、**语用**。④并行化,发展具有合适并行成分的并行语言。⑤可视化。

参考文献

1. 徐家福. 系统程序设计语言. 北京:科学出版社,1983

2. Sammet J E. Programming languages: history and fundamentals. Englewood Cliffs, NJ: Prentice Hall,1969

3. Patt T W. Programming languages: design and implementation. 2nd ed. Englewood Cliffs, NJ: Prentice Hall,1984

4. Tucker A B. Programming languages. 2nd ed. New York: McGraw-Hill,1986

(徐家福)

chengxu yanzheng

程序验证 (verification of programs)　检验程序正确性的理论和方法。为了解一个程序是否正确地实现了预定的目标,传统程序调试方法是通过对程序输入一些规定初始数据,试验性地执行这个程序,测试其是否能产生所要的答案。如果发现有误,就检查和修改所编的程序,直至对所有规定的初始数据,都能产生预期的结果。然而,对于一般程序

而言,它对不同的初始输入数据的加工过程是不同的,而且初始数据的取值范围往往又十分广泛。因此,使用调试方法穷尽程序的各种可能加工过程以确保程序的正确性,几乎是不可能实现的。所以说,调试方法只能发现程序中的错误,多一次测试正确只能说明程序可靠一点,不能说明程序正确无误。程序验证则是研究如何使用数学方法,严格证明一个程序符合其预定目标,因而是正确无误的。

美籍匈牙利科学家 J. von Neumann 在 1947 年发表的论文中就提到程序正确性证明。美国科学家 R. W. Floyd 于 1967 年系统地提出验证程序正确性的归纳断言方法,引起了计算机科学界研究程序验证的热潮。英国科学家 C. A. R. Hoare 于 1969 年将归纳断言方法形式化,提出程序验证的公理系统。1969 年以来又陆续出现很多使用归纳断言方法或者结构归纳法的自动程序验证系统,其中以 20 世纪 70 年代中期实现的 Boyer-Moore 程序验证系统最为著名。1970 年以来还出现能辅助用户正确编制程序的实用的半自动化程序验证系统。在使用这种系统时,用户必须协助系统完成程序验证中创造性最强部分的工作。

为验证程序,必须首先将程序所要实现的目标形式化,即使用数学公式表达程序加工的初始数据的范围(称作输入谓词)和程序加工的结果(称作输出谓词)。程序断言正是对程序性质的描述,形如 $\{\varphi\}S\{\psi\}$,其中 φ,ψ 为两个谓词,S 是一个程序,φ 称为 S 的前置断言,又称输入谓词,ψ 称为后置断言,又称输出谓词。断言 $\{\varphi\}S\{\psi\}$ 称为 S 关于 (φ,ψ) 的正确性断言。它的含义是:若 S 开始执行时 φ 为真,则 S 的执行必终止且终止时 ψ 为真。程序设计的任务就是:对给定的程序规约要求 (φ,ψ),构造程序 S,使得断言 $\{\varphi\}S\{\psi\}$ 为真。反之,程序验证则是对已有的程序 S,验证它是否满足规约 (φ,ψ) 的要求。下面用一个非负整数的除法程序来说明。图 1 中,x_1 是被除数,x_2 是除数,z_1 中存放程序加工后得到的商;z_2 中存放得到的余数;y_1,y_2 是程序加工时使用的工作单元。START 表示程序的起始,HALT 表示程序的终止。方框中是同时赋值语句,如 $(y_1,y_2)=(0,x_1)$,表示将 y_1 置 0 值的同时,将 y_2 的值置为 x_1。圆框内是测试语句,用于控制程序加工的流程。如框图中的语句 $y_1\geqslant x_1$ 表示当 y_1 的值大于等于 x_1 时,程序按 Y 的箭头继续执行,否则按 N 的箭头继续执行。

若约定各个变量的取值都是整数,上述除法程序的输入谓词和输出谓词分别为 $(x_1\geqslant 0)\wedge(x_2\geqslant 2)$ 和 $(x_1=z_1x_2+z_2)\wedge(0\leqslant z_2<x_2)$。

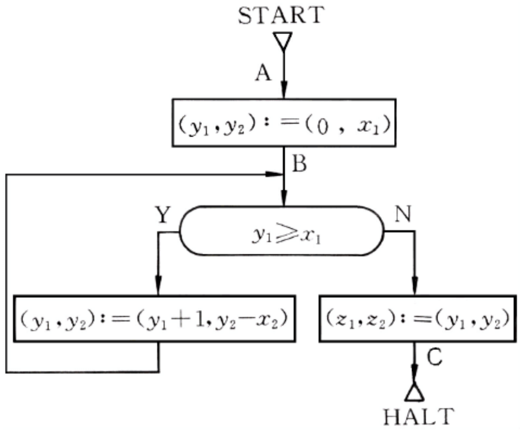


图 1 非负整数的除法框图

在用归纳断言方式证明程序正确性时,还必须在程序的框图中设置一些数学公式,称作断言,表示程序执行到该处时,程序中变量应满足的数学关系。输入谓词可选作起点处的断言,而输出谓词可选作终止点处的断言。

在除法程序中设置三个断言,A 处和 C 处的断言分别为上述输入和输出谓词,B 处断言为

$$(x_1=y_1x_2+y_2)\wedge(y_2\geqslant 0) \quad (1)$$

反映了 y_1,y_2 中存放商数和余数的中间结果值。

验证程序的正确性,就是证明在程序的任何一种可能的加工过程中所设置的断言都是成立的。程序的一个加工过程就是框图中的一个流程。除法程序的所有可能的流程都是由图上的三条路径组合而成:由 A 至 B;由 B 出发回到 B;由 B 至 C。这样,验证程序的正确性,就是证明对任一条路径,只要起点的断言成立,则终止的断言也成立。

以第二条路线为例,它是一条环路。要证明下列命题:若程序执行到环路的起点 B 时,断言(1)成立,则程序执行一周,再达到 B 点时,断言(1)仍然成立。

环行该圈,就是在 $(y_2\geqslant x_2)$ 成立的条件下,执行赋值语句 $(y_1,y_2)=(y_1+1,y_2-x_2)$,而上述语句的执行结果是使 y_1 的取值为执行前 y_1 的值加 1,y_2 的取值为执行前 y_2 的值与 x_2 的差,其他变量的值不变。为保证执行该赋值语句后断言(1)仍然成立,就要求将断言(1)中的 y_1 代为 (y_1+1),y_2 代为 (y_2-x_2) 后得到的公式在执行该语句前成立。即

$$(x_1=(y_1+1)x_2+(y_2-x_2)) \wedge(y_2-x_2\geqslant 0) \quad (2)$$

在执行上述赋值语句前成立。但已知执行该语句前断言(1)和测试条件($y_2 \geq x_2$)均成立。由此推断公式(2)是成立的。这样就完成了对第二条路线的验证,对其余两条路线的验证类似,从而可以证明除法程序的正确性。

归纳断言方法由建立断言和对各条路径逐条验证两部分组成。建立断言是一种创造性的工作,而验证路径的工作尽管繁琐,却是机械的。如何由计算机系统协助用户归纳出合适的断言,是程序验证程序中的重要课题。

用上述方法只能证明在输入谓词成立的前提下,程序终止时输出谓词一定成立。但不能证明在输入谓词成立时,程序一定能终止。不讨论程序终止性的程序验证称为程序部分正确性的验证。包括终止性的验证,则称为程序完全正确性的验证。

程序验证技术除了用于证明程序的正确性,或辅助用于编制正确程序外,还可从程序正确性角度评价程序设计方法和程序设计语言的优劣。但是,保证程序正确性的有效办法,不是在编制程序后再去验证,而是设法在编制过程中,使用适当的技术,使产生的程序正确无误。这类技术叫作程序综合和程序变换。程序验证技术和程序综合变换技术相互参照,共同发展。

参考文献

Manna Z. Mathematical theory of computation. McGraw-Hill,1974

(李晓山　周巢尘)

chouxiang daishu

抽象代数(abstract algebra)　研究各种抽象的代数运算系统的数学分支,又称近世代数。抽象代数是在初等代数基础上,通过对于数系和运算概念的推广,从18世纪萌芽,于20世纪20年代初步成形建立起来的现代数学主要分支之一。

初等代数研究实数或者复数以及以它们为系数的多项式的性质,其中心问题是代数方程组解的计算及其分布的研究。抽象代数是研究文字、向量、矩阵以及更一般的其他抽象对象的代数运算系统。一个代数系统是一个非空的集合,连同定义在这个集合上的一族(可以无穷)抽象运算(即,从多个元素到一个元素的映射),同时有一组公理描述集合元素和运算之间满足的性质。根据这些公理的不同,代数系统被赋予不同的结构,抽象代数的研究内容主要是各种代数系统的结构性质。典型的代数结构有**群**、**环**、**域**、模、代数、**格**、**泛代数**、范畴等。

抽象代数的研究增加了数学与自然科学、社会科学、工程技术等方面逻辑关系的整体认识。现今,在计算机科学领域的所有方面都会用到抽象代数的概念和方法。人们发现,在计算机领域中一些表面看来完全不同的问题,通过数学的表达成为一个抽象代数的系统后,其背后的深刻性质被揭示出来,这对于计算机科学是十分重要和极其有益的。通过抽象代数的语言,人们对于计算机科学的问题有了更加深入的理解与洞察。

群、环、域、模这些系统是早期的和典型的抽象代数系统,是从代数方程组的解的结构的研究逐步发展起来的,突破了单纯从方程求解角度研究问题的局限,开创了从解的结构与形态(几何形式)来研究解的性质的新途径。群、环、域都是具有一个或者两个运算的系统,通过公理化的定义,满足诸如结合律、分配律这些基本的运算律。这些系统既是我们所熟知的一些数学系统的公理化刻画,同时也覆盖了大量的在现实世界中存在的系统。任何现实自然系统与社会系统,只要能够建立起符合这些公理的表述,都可以看成是抽象代数系统,进而借助数学中的表示理论,与已知熟悉的代数系统建立某种关联,可以在极大程度上运用数学方法来研究这些系统。这是人类科学研究的一个公认的有效途径。对于计算机科学而言,也是使用计算机实现问题求解、系统设计和人类行为理解的基本思路和方法。

同态和同构的概念在抽象代数中起核心作用,它是通过一种严格的数学语言,建立起不同代数系统之间的关联。不是局限于单独的代数系统具体结构,而是用统一的观点来看待不同代数系统之间的相似性和类比性。这就提供了一个更高层面的研究代数系统的观点和方法,有益于把一些具体代数系统上的研究结论推广到更多的代数系统上,这是抽象代数形式化研究方法所具有的突出特点,也是计算机科学在研究各种问题时所广泛使用的模式。

把群、环、域以及模这些具体代数系统中共同的概念和平行的结果推广到一般的代数系统,这就产生了**泛代数**的概念。泛代数以各种不同的代数系统之间的共性为主要研究对象,在所有的代数系统中,它具有最广泛的覆盖性和最一般的代表性,自然地,它对于计算机科学中的众多领域,例如自动机理论、程序语言理论、形式语义学、数据库理论等都有深刻的应用。

将一族代数系统以及它们之间的映射作为研究对象,建立相应的代数系统,就会发现许多代数系统

以及非代数系统(包括一些拓扑系统和动力系统)之间的内在联系,这些联系反映了代数学与其他数学分支之间的本质上的共性。在此基础上建立的**范畴论**,正是用于研究这些数学系统中的共性理论。范畴是一种高层次的抽象语言,在代数几何、拓扑学、微分几何等领域都有重要应用。在计算机科学领域中,范畴理论在程序语言学、网络动力学等方面有应用。

由于抽象代数所研究的对象具有很一般的广泛性,方法上以形式化作为研究手段,结论往往具有构造性和算法性的特点,因此与计算机科学具有天然的紧密联系。任何一个系统(自然的、社会的、人造的)都可以在适当定义元素(在计算机科学中,一般称为对象(object))和相应的运算(在计算机科学中,一般称为动作(action))后形成一个代数系统,这个过程称为系统的代数建模。在计算机科学中,通过代数建模来研究系统的内部结构成为常用的手段,并在此基础上开展体系结构、算法设计、软件开发、质量论证、可靠性检验、正确性评价等理论和技术研究。实际上,在计算机科学研究的绝大多数问题中,只有建立了代数模型以后,才能方便地将问题形式化和离散化,真正转化为计算机可以解决的形式。反过来,计算机科学中出现的一些问题也丰富了抽象代数的内容,推进了抽象代数的发展。与计算机科学密切相关的一些抽象代数结构也被提了出来,形成了新的抽象代数研究内容,这些代数结构包括逻辑代数、编码代数、进程代数、语言代数学、**自动机理论**等。

参考文献

1. Jacobson N. Basic algebra, Vol. 1, 2. San Francisco, CA: Freeman Press, 1980

2. Rosen K H. Discrete mathematics and its applications. 7th ed. New York: McGraw-Hill, 2011.(罗森. 离散数学及其应用. 原书第6版. 袁崇义,译. 北京:机械工业出版社,2011)

3. Burris S, Sankappanavar H. A course in universal algebra. New York: Springer-Verlag, 1981

(李廉)

chouxiang shuju leixing

抽象数据类型(abstract data type) 与表示无关的**数据类型**。数据类型由一个对象集合(值集)和在该集合上定义的若干合法运算所组成的运算集合组成。抽象数据类型用数学方法定义对象集合和运算集合,仅通过运算的性质刻画数据对象,而独立于计算机中可能的表示方法。其目的在于隐蔽运算实现细节和内部数据结构,同时向用户提供该数据类型的完整信息。

抽象数据类型的概念是逐步形成的。20世纪60年代末到70年代,人们将用户自定义的类型称作抽象数据类型。传统的算法语言没有用户自定义类型的设施。到60年代末,为了实现算法细节和数据内部结构的隐蔽,SIMULA 67语言中引入了**类**,随后出现了模块概念。模块可分成模块式和模块体,模块式定义外部可见的运算接口,模块体对外不可见,其中可定义私有的数据结构和运算。通过接口和实现的分离,模块提供了用户自定义类型的手段,达到数据抽象、信息隐蔽的目的。在这个意义下,模块所定义的数据类型称作抽象数据类型。实际上,模块提供了一种抽象数据类型实现的手段。这种手段在Modula-2,Ada等语言中得到进一步的完善和发展。用户自定义数据类型的另一优点是设计与实现相分离,可将模块式看作设计规约,而模块体是相应的实现。这种分离推动了软件规约的研究,也进一步推动了抽象数据类型的研究。

当用一个数据类型去模拟一类客观对象时,可先给出该类型的性质和功能的描述,然后用已有的语言设施和数据类型实现所需的功能,并证明实现的正确性。仅通过模块式描述数据类型的型构是不够的,还必须用抽象的方法完整地描述对象的性态和功能。这就是用抽象数据类型表示功能规约。这时,抽象数据类型完全独立于具体表示,反映出纯抽象的性质。抽象数据类型的规约方法主要有二:其一是代数方法;其二是模型方法。代数方法基于G. Birkhoff,J. D. Lipson等的异调代数理论,经S. Zills,J. A. Guttag等人的发展,其理论基础日趋完善,并逐步应用于**软件工程**实践,成为有代表性的抽象数据类型规约方法。模型方法基于C. A. R. Hoare的**前后断言方法**,它通过已定义的(抽象)数据类型来给出所要定义的新类型的抽象模型。

采用代数方法,抽象数据类型的规约由两部分组成,一是**语法**部分,二是公理部分。语法部分给出了抽象数据类型的名及其上运算的定义域和值域,公理部分则通过给出一组刻画各运算之间相互关系的方程来定义各运算的含义。从**语义**的角度,代数规约的语义是一类代数。在语法正确的基础上,语义正确性是指相应代数满足规约中公理部分的所有公理。通常,具语义正确性的语义模型代数有多个且形成一个谱,谱的两端分别称为始语义模型和终

语义模型。该谱系为实现者提供了较大的灵活性。基于代数方法的规约语言有 OBJ,Clear 等。例如用 OBJ 写的栈定义如下:

```
obj stack;
sort stack /integer,boolean;
ok_ops
        push: stack,integer → stack;
        pop:  stack → stack;
        top:  stack → integer;
        empty:stack → boolean;
        newstack: → stack;
        depth:stack → integer;hidden;
error-ops
        underflow → stack;
        no_more → integer;
        overflow → stack;
op-equ's
        pop(push(s,item)) = s;
        top(push(s,item)) = item;
        empty(newstack) = true;
        empty(push(s,item)) = false;
        depth(newstack) = 0;
        depth(push(s,item)) = 1 + depth(s);
error-equ's
        pop(newstack) = underflow;
        top(newstack) = no_more;
        push(s, item) = overflow if depth(s)
        >100;
job
```

模型方法不是对抽象数据类型中运算的性质加以直接刻画,而是通过某些基本类型和已定义的(抽象)数据类型来给出所要定义的新类型的抽象模型,用已定义的运算的性质来间接刻画要定义的数据类型中运算的特性。一般说来,抽象数据类型的模型规约由以下三部分组成:①状态集的定义(可能包含不变式);②初始状态定义(通常只有一个);③运算集的定义,通常采用输入输出断言刻画。值得注意的是,模型只能理解成行为的描述而与具体实现无关。但如果不加注意,模型规约可能引入与实现有关的内容。由于这种方法与正确性证明方法有较直接的对应,因此,在规约语言中得到了广泛的应用。如 Z,VDM 等。

抽象数据类型能够从抽象的角度描述客观对象的性态,满足信息隐蔽、功能抽象、设计在前、方便验证等软件工程的要求。其理论对软件规约、正确性证明等课题的研究均具有重要意义,对**面向对象语言**的发展产生了重要的影响。

参考文献

1. Liskov B H, Guttag J. Abstraction and specification in program development. Cambridge, MA: The MIT Press, 1986

2. Guttag J V. Notes on type abstraction. In: Proc. Specification of Reliable Software Conf., Cambridge, 1979

(伊波 吕建)

chufaqi

除法器(divider) 对以数字形式表示的两个 n 位数求商的一种运算电路。

两个原码数相除,其商的数值为两数绝对值相除后的结果,而符号为两数符号的异或值。设 x、y 为被除数与除数,x_s、y_s 为被除数与除数的符号,则其商 $q=|x|/|y|$,符号 $q_s=x_s \oplus y_s$。

除法运算虽然使用频度低,但它是基本的四则运算之一,因此大多数计算机有除法器。与乘法的处理思想相似,除法运算的常规算法是将除法转换成若干次"加减-移位"循环。常见的除法算法有恢复余数法和不恢复余数法。恢复余数法是,不管被除数(或余数)减除数是否够减,都一律做减法。若余数为正,表示够减,该位商上"1";若余数为负,表示不够减,该位商上"0",并要恢复原来的被除数(或余数)。恢复余数法一般很少采用,原因是恢复余数时浪费一次加法时间,同时造成除法进行过程的步数不固定,控制起来比较复杂。最简单并被广泛采用的除法器是不恢复余数法除法器。

二进制原码不恢复余数除法的规则可由下式表示

$$r_{i+1} = 2r_i + (1 - 2q_i) \times y$$

式中,r_i 和 r_{i+1} 分别为第 i 次和第 $i+1$ 次求商所得的余数,q_i 为第 i 步所得的商,即当余数为正时,上商为"1",余数左移 1 位,下一步减除数;当余数为负时,上商为"0",余数左移 1 位,下一步加除数。由于加减运算交替进行,故也称为原码加减交替法。

现举例说明原码加减交替除法的实现。设 $x=0.10101$,$y=0.11110$,则

$$\frac{x}{y}=\frac{0.10101}{0.11110}$$

$|x|=0.10101 \to$ A(寄存器),$|y|=0.11110 \to$ B(寄存器)

$[-B]_{补}=1.00010$,$0 \to$ C(寄存器)

运算过程如下:

	A	C	
	00.10101	0.00000	
$+[-B]_{补}$	11.00010		$-\lvert y\rvert$
	11.10111	0.00000 0	余数为负，商0
←	11.01110		左移一位
+B	00.11110		$+\lvert y\rvert$
	00.01100	0.00001	余数为正，商1
	00.11000		左移一位
$+[-B]_{补}$	11.00010		$-\lvert y\rvert$
	11.11010	0.00010	余数为负，商0
←	11.10100		左移一位
+B	00.11110		$+\lvert y\rvert$
	00.10010	0.00101	余数为正，商1
←	01.00100		左移一位
$+[-B]_{补}$	11.00010		$-\lvert y\rvert$
	00.00110	0.01011	余数为正，商1
←	00.01100		左移一位
$+[-B]_{补}$	11.00010		$-\lvert y\rvert$
	11.01110	0.10110	余数为负，商0
+B	00.11110		
	00.01100		最后一步恢复余数，$+\lvert y\rvert$

又由 $q_s = x_s \oplus y_s = 0 \oplus 0 = 0$，得 $\frac{x}{y} = 0.10110 + \frac{0.01100 \times 2^{-5}}{0.11110}$。

实现原码加减交替除法器的算法流程如图 1 所示，其中三个寄存器 A、B、C 分别存放被除数、除数和商，计数器 CR 用来控制移位次数，C_0 为 C 寄存器的最低位。

需要注意的是，在定点小数除法运算时，为了防止溢出，要求被除数的绝对值小于除数的绝对值，且除数不能为 0。另外，在原码加减交替法中，当最终余数为负数时，必须恢复余数，使之变为真余数，但此时不能再左移了。

为了提高除法的运算速度，可采用跳 0 跳 1 除法、阵列除法器或迭代除法。跳 0 跳 1 除法的原理是：当前一步除法所得余数很小，即可连商多位，余数前面的 0 越多，连商的位数就越多，这将减少除法的步数，从而提高除法的速度。阵列除法器是利用若干个加减单元组成阵列，将各步"加减-移位"操作在一步内完成。迭代除法的基本思想是通过多次乘法运算来获得商，以加快求商的速度。

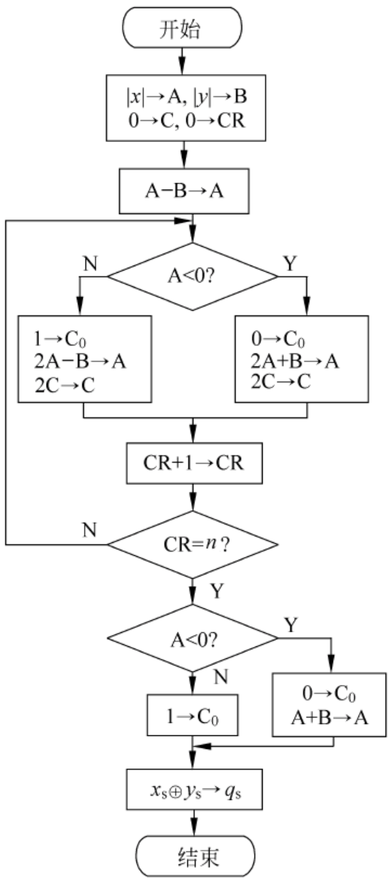


图 1　原码加减交替除法流程

参考文献

1. 蒋本珊. 电子计算机组成原理. 北京：北京

理工大学出版社,1994

2. 王爱英. 计算机组成与结构. 3 版. 北京:清华大学出版社,2001 (刘恩德)

chuliji(qi) tixi jiegou

处理机(器)体系结构(processor architecture) 从汇编语言程序设计员的角度所看到的处理机的内部功能结构。处理机体系结构以外,还有计算机体系结构,后者不仅包括处理机,还包括存储器和外围设备。

处理机体系结构是围绕着计算机指令系统的执行来设计的。一条典型指令的执行包括以下阶段:取指令,指令译码,计算存放操作数的有效地址,取操作数,执行指令所规定的操作以及送回操作结果。不同的指令在各个阶段的操作不同,控制指令的操作是处理机中控制器的职能;执行指令的算术运算或逻辑操作的部件是处理机的运算器;在处理机中暂时存放指令、控制字、源操作数、中间结果和最后操作结果的是处理机中的寄存器。控制器、运算器和寄存器是构成处理机体系结构的基本部件。在新型处理机体系结构中,指令的执行也可以是并发地执行的,这需要有流水线执行部件和相关控制部件,指令发射和调度部件,有的还有指令转移预测部件;为提高存取寄存器以外的操作数,片上可以设有高速缓冲存储器(cache)和多级 cache 以及相应的存储管理部件等等。

处理机又分向量处理机(vector processor)和标量处理机(scalar processor)。向量处理机的**指令系统**包含大量专门处理一维数组的数据(这种数据称向量)的指令;向量处理机体系结构中,有较大数量的寄存器,便于寄存大量数组数据并快速读出和写入;而大多数处理机是标量处理机,它主要处理一般单个数据。

处理机体系结构分类

1. 从指令执行和复杂程度不同,处理机体系结构可分成以下几种:

(1) 单发射(single issue)指令和多发射(multiple issue)指令结构 在单一个机器周期(参见**指令周期**)内,处理机只发射一条指令,并只执行一条指令者,为单发射结构;如果在一个机器周期内,发射多条指令,并可以执行多条指令者,为多发射结构。如奔腾 4 处理机,可以在一个周期内发射三条指令,并有两条浮点流水线可以同时执行两条浮点指令。

(2) 指令的顺序和乱序 为提高每个机器周期可以并发执行指令的数目,指令可以顺序和乱序执行,这样,又可以分成顺序执行(in order execution)和乱序执行(out of order execution)结构。过去传统的处理机体系结构中,指令的执行是按照从指令存储器中取出的指令流的顺序来执行的,称为顺序执行结构。为提高指令执行并行度,扩大可以进行指令调度的基本块,把指令存储器中取出的指令流放在一个指令缓冲器中,并根据操作数是否已经即时可用,进行调度,打乱原来指令缓冲器中的指令流的顺序,乱序来执行,称为乱序执行结构。当然,由于指令乱序执行,需要保存过程中的一些执行结果,并进行整理,才最终把结果写入寄存器堆。如 Intel 公司在奔腾 I 以前生产的处理机是顺序执行体系结构,而奔腾 II 以后的处理机都是乱序执行体系结构。

(3) 由于计算机的指令系统的复杂程度不同,又可以分成**精简指令集计算机**(RISC)结构和**复杂指令集计算机**(CISC)结构 CISC 的特点是用微程序存储器来管理指令的执行,一条指令由存放在微程序存储器中的多条微指令组成,而一条指令的执行周期由多个微指令执行周期组成。CISC 的指令长度是可变的。而 RISC 的特点是指令系统中尽量多采用一个机器周期内可以执行完毕的寄存器操作指令,控制指令执行的是硬布线的控制器,RISC 指令执行效率较高。现代实际上的处理机,往往是 CISC 和 RISC 机构的混合。

2. 从片上"核"(core)的数量不同,处理机体系结构可以分成多核(multi-core)芯片、众核(many core)芯片和带协处理器的异构多核(heterogeneous multi-core)芯片等处理机体系结构。核是处理机中取指令和执行指令的部件。过去,为提高处理机芯片的性能,芯片设计者用提高核的工作频率来实现;而芯片工作频率过高,会使芯片的功耗太大而无法实现。同时,VLSI 工艺的发展使得芯片上可以容纳的晶体管数目不断增加,于是芯片设计者把多个核做在同一芯片上,使几条指令可以在芯片上并行执行,以提高处理机芯片的功能。芯片上含有多个核者称为多核芯片。如 AMD Phenom II X2 和 Intel Core Duo 为双核芯片,AMD Phenom II X6 和 Intel Core i7 Extreme Edition 980X 为六核芯片。多核芯片的指令并行执行是采用多机技术,如共享总线的对称多处理机(SMP)技术。如果芯片上的核数目进一步增加,由于数据操作和指令操作中发生的资源竞争等问题而用一般传统的多处理技术没有办法解

决时,需要设计片上的网络(network on chip)以连接众多个核,这称为众核芯片。众核芯片上核的数量一般要到几十个以上。

3. 从应用不同方面可以对处理机体系结构分类,其中比较常见的有**图形处理机(器)**(GPU)、数字信号处理机(器)(DSP)和**网络处理机(器)**等。图形处理机(器)GPU 又称可视化处理部件 VPU,它是一种专用处理机(器),用来加速图形处理的。由于处理图形的算法比较适合高度并行地执行,所以 GPU 可以由很多个同构的处理图形的部件组成,处理图形速度很快。它应用在嵌入式系统、移动电话、个人计算机和工作站等。比较早问世的 GPU 是 NVIDIA 公司发布 GeForce 256。数字信号处理机 DSP 是对于快速处理数字信号具有优化体系结构的处理机。数字信号处理要求对于一组数据进行大量的重复计算,其算法包含大量的加法和乘法,所以实现的部件比较简单而且可以用较大量同构的部件高度并行执行。网络处理机是专门用于现在网络和通信协议的处理,在硬件设计中多采用多核并行处理,以提高速度;在软件设计中通常是允许可编程的,以满足不同类型的需要。网络处理机应用在路由器和网络监控器中。

处理机体系结构发展方向

处理机体系结构的发展是和 VLSI 工艺的发展分不开的,其发展方向是提高性能和降低功耗。由于 VLSI 工艺继续不断地进展,芯片上的晶体管数目还在增长,但是,单靠用提高芯片上电路的工作频率而提高处理机性能是行不通的,这受到功耗发热的限制。因此,多核芯片和众核芯片是提高处理机性能的重要途径。另外,为了提高工作速度,在多核芯片上增加结构简单而对于某种应用速度很快的加速协处理机核,即异构的众核芯片,也是重要途径。然而,无论是多核、众核和异构多核的处理机体系结构,要充分挖掘和发挥芯片上各个核的工作并行性,还要靠编译来解决。软件方面的底层并行编译是当前需要解决的一个难题。此外,要提高性能,不仅众核的运算速度要快,它们访问片上的高速缓存来读数和存数的速度也必须快;还有众核之间的片上网络连接要效率高;所以**片上系统** SOC 的要求也愈来愈高、愈来愈复杂,实际上是把过去传统的计算机系统技术放在芯片上实现;这些都是目前研究处理机体系结构的重要课题。近年来,移动终端和三代、四代手机应用愈来愈广泛,因此,移动终端和手机上的处理机体系结构也是研究的重要方向。这个领域里,低功耗研究的重要性尤为突出。

参考文献

1. 李三立. 微处理器与微计算机. 北京:国防工业出版社,1981

2. Hennessy J L, Patterson D A. Computer architecture: a quantitative approach. 5th ed. Morgan Kaufman Publishers Inc. , 2011

3. 李三立,李亚民. RISC——单发射与多发射体系结构. 北京:清华大学出版社,1994

(李三立)

chuliqi guanli chengxu

处理器管理程序(processor manager) 参见**任务调度程序**。

chuping

触屏(touch screen) 一种用手指或笔触及屏幕上所显示的选项来完成指定的工作的人机交互式**输入设备**。它使用方便,反应灵敏,得到广泛应用。

触屏在安装方式上有两种:一种是外挂式,把触屏附装在荧光屏外面;另一种是内装式,把触屏在生产显示器时装配成一整体。

触屏由三个部分组成。一是传感器,把人的手指或笔触及的位置检测出来。二是控制卡,位置信号经过模数转换器,形成数字信号,经接口送入计算机。三是驱动程序,即相应的管理软件。

传感器的主要类型:

(1)电阻膜触屏 电阻膜触屏的屏体部分是一块多层复合板。有机玻璃作为基板,表面上涂有一层透明的导电层,基板上是内表面也涂有导电层的塑料薄膜,在两层导电层之间有许多细小的绝缘物隔开。导电层间加有电压。当手指接触屏幕时,导电层间出现一个接触点,电阻发生变化,即可得触摸点的 Y 轴坐标和 X 轴坐标。进行数模转换后送入计算机。电阻膜传感器的缺点是透光性差,只有 70% ~80% 的透光度,因而影响了显示器屏幕的亮度。电阻触屏的寿命不长。它的优点是响应速度快,分辨率高(可达 4096 点 ×4096 点),而且价廉。

(2)电容触屏 电容触屏的构造是一块镀有一层透明导电薄膜的玻璃板。装配时导电薄膜朝向显示器屏幕。屏体的四个角或四条边上引出四个电极,在触屏内形成一个低电压交流电场。当触摸屏幕时,人的手指与导电层间会形成一个耦合电容,四个电极间的电容也就发生变化。这个变化送到控制

器,便可计算出触摸点的位置。电容触屏的双玻璃设计能保护导体层及感应器,寿命长。

(3) 红外线触屏　红外线传感器是在屏幕的上下和左右一一对应地装上红外发光管和红外光敏元件,在屏幕表面形成一个红外线网。当触及屏幕某一点时,便会挡住经过该点的横竖两条红外线,可即时算出触摸点位置。分辨率可达 4096 点 ×4096 点(2011 年)。红外线触屏抗干扰能力强,适宜某些恶劣的环境条件下使用;而且价格低廉,安装方便。

(4) 表面声波触屏　表面声波触屏是一块玻璃平板的左上角和右下角各固定了竖直和水平方向的超声波发射器,右上角和左下角安装了两个相应的超声波接收器。当手指或其他能够吸收或阻挡声波能量的物体触摸屏幕时,声波能量被部分吸收,接收到的波形有衰减或时间变化,通过计算即得触点坐标。表面声波触屏还能响应触摸压力大小,不受温度、湿度等环境因素影响,分辨率可达 4096 点 × 4096 点(2011 年),寿命长,适合公共场所使用。

触屏除了在计算机中应用外,也应用于手机。特别是在信息查询上,在商业、金融、交通、旅游等领域,得到极其广泛的应用。

参考文献

程子华,阳胜峰. 触摸屏应用技术. 北京:人民邮电出版社,2010

（林 兼）

chuangan shoutao

传感手套(sensor glove)　用于捕获用户手部动作的、像手套一样戴在手上的人机交互输入设备(又称***数据手套***)。通常用于虚拟环境中的人机交互操作,目前已在某些应用领域(如数字产品评价、三维游戏)中得到了应用。

顾名思义,传感手套中附有传感器,它们分布于手掌和手指的关节处,用于获取用户手形准确的配置信息并借助于数据采集装置将其转换为数字信号。通常,在使用传感手套时,还需要在用户的手腕部或手背上放置一个 6 自由度定位跟踪器(例如,电磁跟踪器、惯性跟踪器或光学跟踪器)以跟踪用户手的整体方位。一旦计算机接收到用户手部动作的完整信息,即可启动处理程序解算用户的手部运动意图,以控制人机交互过程或提供准确的视、听、触力觉反馈,使用户手部运动能真正迁移到虚拟环境,从而便利用户与虚拟环境中虚拟物体的直观自然的交互。

传感手套的关键是传感器技术。不同种类的传感手套中的传感器并不一样,有的采用电阻墨水传感器,有的采用光纤传感器,还有的采用张力测量弯曲传感器等,但它们都需要能够实时反映用户的手掌和(或)手指的运动,将借助于数据采集装置将其转换成关节角度数据。通常,嵌在传感手套中的传感器很薄、很柔软,不会让用户产生手套中有异物的感觉。传感手套中传感器的数量可多可少,主要取决于其配置方式。最常见的传感器配置数目为 18 和 22 两种。如图 1 所示,22 个传感器的位置如下:大拇指掌指关节和近端指间关节处各 1 个;食指、中指、无名指和小指的掌指关节、近端指间关节和远端指间关节处各 1 个;每两个相邻手指之间的指间关节处各 1 个;手掌中 2 个;手腕上 2 个。而 18 个传感器的配置中,只是在除大拇指外的其余 4 个手指的远端指间关节处各减少了 1 个。



图 1　传感手套的 22 个传感器配置

商用传感手套的种类较多。低端的传感手套仅能测度两个或多个手指之间是否发生接触,或仅内嵌数个弯曲传感器;常规的传感手套一般具有 18 个(或 22 个)弯曲传感器;而高端的则可通过在传感手套外附加振动装置或外骨骼(exoskeleton)结构,为用户手指提供触、力觉反馈,在此意义下,传感手套亦可用作输出设备。

计算机视觉技术的快速发展使得通过摄像机捕获用户手部精细动作成为可能。通过在用户手指尖佩戴轻量级标记物,采用立体视觉技术实时跟踪标记物空间位置,借助逆运动学解算各手指弯曲角度,可为传统的传感手套提供一种可行的替代。

传感手套有左、右手之分,但目前虚拟现实应用

中真正采用双手操作的还很少。

参考文献

1. Zimmerman T G, Lanier J, Blanchard C, et al. A hand gesture interface device. In: Proceedings of the SIGCHI/GI Conference on Human Factors in Computing Systems and Graphics Interface. ACM Press, 1987, 189-192

2. Sturman D J, Zeltzer D. A survey of glove-based input. IEEE Computer Graphics and Applications. 1994. 14(1): 30-39 （万华根）

chuanganwang

传感网(sensor network) 泛指采用有线或无线通信技术将内置传感器的设备组织成网络,用于监测周围环境中的物质运动现象。传感网扩展了传统计算机网络中单一的人与人信息交互模式,有能力提供人与物、物与物之间的信息交互,将网络末梢延伸与拓展到了更为广阔和深入的物理世界。传感网是一个较为宽泛的技术概念,广义的传感网有多种形态,主要包括无线传感器网络和工业测控网络等。

无线传感器网络(wireless sensor networks)出现于20世纪90年代初,得到计算机、网络、微电子、传感器、通信、信号处理等多学科多领域的普遍关注。无线传感器网络一般指集成有传感器、数据处理单元、通信模块和电池等功能模块的微小节点通过自组织的方式构成的网络,其网络的主要特征是无线、大规模、自组织、多跳、无分区、无基础设施支持,而其中的网络节点通常是同构的、成本较低、体积较小、大部分节点无法移动、被随意撒布在工作区域。为适应在无基础设施支持的恶劣环境下长时间工作的需要,低功耗成为无线传感器网络体系结构和协议设计的主要技术目标。研究实践表明,无线传感器网络具有突破传统技术限制的非凡潜力,辅助完成了许多令人兴奋的科学研究与试验,取得了意想不到的研究成果,因而被期待在环境观测、医疗健康、结构监测、工业自动化、智能家居和军事等众多领域能取得可以预期的规模化商业应用。无线传感器网络与传统网络技术有着不完全相同的设计目标,前者以数据为中心,而后者却以传输数据为目的。传统网络的设计一般遵循"端到端"的边缘论思想,强调将一切与功能相关的处理都放在网络的端系统上,网络节点仅仅负责数据分组的转发,而在以数据为中心的无线传感器网络中节点上数据的处理和融合通常被认为是合理的选择。此外,虽然无线传感器网络中存在众多一般性的共性技术,但其应用系统的设计往往没有通用的模式,需要根据工作环境和业务特征加以定制与优化。在研究和发展过程中,无线传感器网络自身又衍生出了若干具有一般性的特殊形态,它们继承了无线传感器网络的共性特征,又有各自特有的技术内涵与外延。典型的有水声传感器网络、无线传感器和执行机构网络、多媒体传感器网络和军用雷达网络等。

*测控网络*是工业自动化领域的专用术语。通常指采用各种现场总线技术,如PROFIBUS、CAN、Foundation Fieldbus和LonWorks等,将工业生产线上的智能传感器、数字或模拟I/O设备等连接成网络,形成**集散控制系统**。因测控网络也表现出将传感器互连成网的特性,它也是广义传感网的一种常见形态,并且得到成熟而广泛的商业应用。

技术的发展与进步不断孕育出一些外延更为宽泛的概念,如泛在网和物联网等。传感网常被定义为它们的一种存在形态。

参考文献

1. 任丰原, 林闯. 无线传感器网络的现状与发展. 2005年中国计算机科学技术发展报告,北京:清华大学出版社, 2005

2. 阳宪惠. 现场总线技术及其应用. 北京:清华大学出版社, 2008 （任丰原）

chuanshu kongzhi xieyi

传输控制协议(transmission control protocol, TCP) Internet定义在两台计算机之间进行可靠数据传输的协议。TCP在协议层次结构中位于IP(参见**网际协议**)层之上。TCP允许一台计算机上的多个应用程序同时进行通信,也能对接收到的数据进行分解,分别送到多个应用程序。TCP使用协议端口号来标识一台计算机上的不同目的进程,每个端口都用一个小于65 535的整数来表示。

由于TCP是建立在连接的抽象概念上的,它所标识的对象不是某个端口,而是一个虚电路连接。TCP使用连接而不是协议端口作为基本的抽象概念,连接是用一对端点来标识的。

TCP将端点定义为一对整数(host, port),其中host是主机的IP地址,port是该主机上的TCP端口号。由于TCP使用端点来识别连接,这样一台计算机上的某个TCP端口号可以被多个连接所共享。因此,程序员能设计同时为多个连接服务的程序,而不需要为每个连接设置各自的本地端口号。

TCP是一个面向连接的协议。这种连接采用**客户-服务器模式**建立,即由客户方的应用程序主动请求,服务方应用程序收到客户方的请求后向客户方返回是否同意建立一个连接的响应,客户方要对服务方的响应返回一个确认,因此这个连接建立过程称为三次握手。对于服务方的应用程序来说,这个过程是被动打开的过程。连接建立之后,应用程序开始传输数据。

两台计算机上的TCP软件之间传输的数据单元称报文段。通过报文段的交互可建立连接、传输数据、发出确认、通告窗口大小以及关闭连接。

TCP使用专门的滑动窗口机制来解决传输差错控制和流量控制这两个问题。TCP允许随时改变窗口大小。在每个确认中,除了指出已经收到的分组外,还包括了一个窗口通告,用来说明接收方还能再接收多少数据。可以将窗口通告的值当做当前的接收缓冲区大小。通告值增加,发送方扩大发送窗口;通告值减少,发送方缩小发送窗口。采用这种滑动窗口机制,不仅提供了可靠的传输服务,而且还提供了流量控制功能。但是TCP采用的滑动窗口机制只解决了端到端的流量控制,并未解决整个网络的拥塞控制。由于TCP基于这个滑动窗口对报文段进行编号,因此发现可能的报文丢失、重复或乱序,并使用超时和重传机制进行恢复。

TCP的连接被视为两个单向的传输,任何一方都可以单独提出关闭请求以结束自己的数据发送。如果双方都提出了关闭请求,整个TCP连接才关闭。

参考文献

1. 胡道元. 计算机网络. 2版. 北京:清华大学出版社,2009

2. Comer D E. Internetworking with TCP/IP, Volume Ⅰ. 4th ed. Engliwood Cliffs, NJ: Prentice Hall, 2000

(胡道元)

chuanshuwang jiekou

传输网接口(transmission network interface)　整个电信网按网络功能分为三个部分:传输网、交换网和接入网。为保证通信系统之间的互联互通,按照网络与网络之间接口关系定义为业务节点接口(SNI)、用户网络接口(UNI)、网络与网络接口(NNI)。这些接口的技术要求需要予以标准化。

传输网作为现代通信网的基础网络,按照覆盖地域的不同,可分为国际传输网与国内传输网,按照传输距离划分又可分为长途传输网与本地传输网。网络与网络接口(NNI)定义了两个网络之间互联时交换信令和管理信息的接口要求。

业务节点(SN)是提供业务的实体,是一种可以接入各种交换型或永久连接型电信业务的网络单元,例如本地交换机、IP路由器、租用线业务节点或特定配置情况下的视频点播和广播电视业务节点等。SNI即是AN与SN之间的接口。

一个接入网(AN)可以与多个SN相连,这样一个AN既可以接入多个分别支持特定业务的SN,也可以接入多个支持相同业务的SN。一个AN可以实现的UNI和SNI的类型和数量原则上没有限制。必须注意,UNI与SN的关联是静态的,即关联的确立是通过与相关SN的协调指配功能来完成的,对SN接入承载能力的分配也是通过指配功能来完成的。

接入网负责将电信业务透明传送到用户,具体而言,接入网即为本地交换机与用户之间的连接部分,通常包括用户线传输系统、复用设备、交叉连接设备以及用户/网络接口。接入网有三种主要接口,即用户网络接口UNI、业务节点接口SNI和Q3管理接口。国际电信联盟(ITU)在1995年发布的G.902标准中对接入网的定义为:接入网是由业务节点接口(SNI)和相关的用户网络接口(UNI)之间的一系列传输实体(包括线路设备和传输设备)组成,它是一个为传输电信业务提供所需传输承载能力的实施系统,可经由Q3接口进行配置和管理。

接入网是由一系列实体(诸如线缆装置、传输设备等)组成的,为在一个业务节点接口(SNI)和每一个与之相关联的用户网络接口(UNI)之间提供电信业务而提供所需传送承载能力的一个实现。UNI、SNI和NNI的接口边界如图1所示。

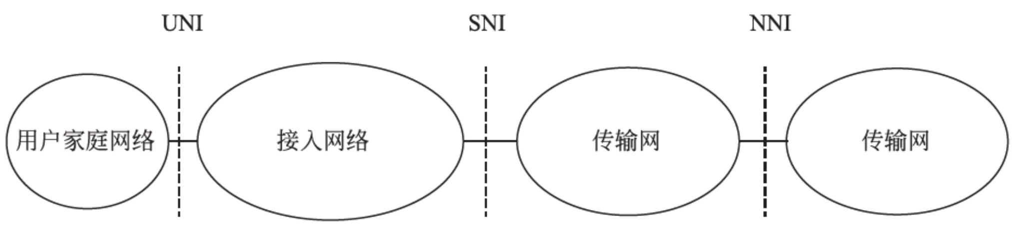


图1　UNI、SNI和NNI的接口边界

参考文献

1. 通信行业标准:自动交换光网络(ASON)技术要求 第1部分:体系结构与总体要求. GB/T 21645.1-2008

2. 通信行业标准:数据通信名词术语. YD/T 1133—2001. 2001

3. ITU-T 标准. G.984.1:SERIES G:Transmission systems and media, digital systems and networks. 2003

(程时端 马严)

chuanshu wumalü

传输误码率(error rate of transmission) 在一定时间内收到的数字信号中发生差错的比特数与其同一时间所收到的数字信号的总比特数之比。误码率是衡量数据在规定时间内数据传输精确性的指标。数据通信中的误码率定义为

$$传输误码率 = \frac{传输中的误码数}{所传输的总码数} \times 100\%$$

对于一个实际的数据传输系统,不能笼统地说误码率越低越好,要根据实际传输要求提出误码率要求。差错的出现具有随机性,在实际测量一个数据传输系统时,只有被测量的传输二进制比特数越大,才会越接近于真正的误码率值。如果传输的不是二进制比特,要折合成二进制比特来计算。IEEE 802.3标准为1000Base-T网络制定的可接受的最高限度误码率为10^{-10}。这个误码率标准是针对脉冲振幅调制(PAM-5),即千兆以太网的编码方式而设定的。

误码的产生是由于在信号传输中,衰变改变了信号的电压,致使信号在传输中遭到破坏,产生误码。噪声、交流电或闪电造成的脉冲、传输设备故障及其他因素都会导致误码。各种不同规格的设备,均有严格的误码率定义,如通常视/音频双向光端机的传输误码率应$\leqslant 10^{-9}$。

参考文献

姚万生. 计算机网络原理与技术. 哈尔滨:哈尔滨工业大学出版社,2007

(周杰)

chuan chuli yuyan

串处理语言(string processing language) 一类用于描述串处理的语言,即一类字符处理的语言。在程序设计语言中,“串”这一名词通常指字符序列,例如,ABC是三个字符组成的串。大多数情况下计算机的输入形式是字符串的形式(例如在终端上键入的命令),同样,计算机的输出大多数也是串的形式。

常规程序设计语言的主要职能集中于数值计算和数据处理,然而,其中总要含有许多重要的串处理工作。例如,编译程序是以字符串作为输入,分析该字符串并生成二进制机器码或字符串输出。命令解释程序也总是分析命令串,并且执行相应的操作。这类程序是经常且广泛使用的程序,显然该类程序必须是高效的。正是基于这一原因,该类程序一般均采用系统程序设计语言来书写,如**C语言**等,而非高级串处理语言。但是,高级串处理语言对诸多复杂问题的处理提供了良好的支持,如:语言转换、计算语言学、计算机代数、正文编辑以及格式化文档生成等。

1957—1958年由V. Yngve设计的Comit为第一个串处理语言,随后出现多种串处理语言,例如:SNOBOL,Ambit,Axle,Convert,Panon,Icon等。

关于数值计算中的数学表示,至今在串处理中既未对其应具有的操作形成统一的认识,也未对串处理有一个标准的记号表示,串处理语言的开发人员几乎是在无先例的情况下开始工作的。因此,串处理语言的记号,程序结构以及对问题的表示研究均与传统的程序设计语言有本质的不同。

目前,串处理中有4种基本操作已广为接受:并置、子串的识别、模式匹配以及串的转换。并置是在一个串之后添加另一个串来构成一个长的串的操作。因此,对串AB和CDE进行并置的结果是ABCDE。这一操作是将串看作是字符序列这一概念的自然扩展。一个串称为子串的定义是它完全含在另一个串中。例如:BC和CDE均是ABCDE的子串。

最重要的串操作是模式匹配:检查一个串,在其中确定对子串的定位,并确定该串是否具有某种性质。例如:确定一个特定的子串的出现,出现的位置,以及子串间的特定关系等。串的转换同模式匹配息息相关,它通过一个串进行模式匹配的结果,再形成对子串的替换。

最流行的串处理语言为SNOBOL。

参考文献

Griswold R E, Poage J F, Polonsky I P. The SNOBOL4 programming language. Prentic-Hall Inc., 1971

(郑国梁)

chuangkou xitong

窗口系统(window system) 通过窗口来控制

计算机显示器及输入设备的一种系统软件。**窗口**是指显示屏幕上的一块矩形区域,它显示用户或系统某一进程的输出。窗口又可看作是一种虚拟终端,一个屏幕上可有多个窗口,显示多个输出,不同窗口可互相重叠。窗口系统所管理的资源有常用的窗口、图符、选单、指点设备,即 WIMP,还有屏幕、像素映象、色彩表、字体及光标等。窗口系统是图形用户界面的基础,它为用户与计算机对话提供了窗口界面、编程接口及窗口管理接口。20 世纪 80 年代初美国施乐(Xerox)公司 Alto 计算机上运行的 Smalltalk-80 程序设计环境是第一个实用的多窗口系统;苹果(Apple)公司的 Macintosh 微型计算机最早在操作系统上使用窗口界面;微软(Microsoft)公司的 Windows 是 PC 机上最重要的窗口系统;麻省理工学院(MIT)等开发的 X 窗口系统广泛应用于各类计算机,其版本 X.11 已成为事实上的工业标准。

窗口系统按内部结构可分为基于核心及基于客户-服务器两类。前者把窗口系统核心放在操作系统内,其运行效率高,但可移植性差;后者把窗口系统核心及窗口应用程序均作为操作系统的用户进程予以运行及相互通信,它具有网络透明、易扩充、易移植等特点。图 1 为基于客户-服务器窗口系统的结构层次图,其中窗口系统核心为服务器进程,用来对显示器及输入设备进行操作;应用程序、终端仿真、窗口管理均作为客户进程通过网络上进程间通信协议与服务器进程通信;语言接口、工具箱及专用界面工具集分别向程序员提供低级、高级及具有某种风格的编程接口。窗口系统的产生极大地影响了操作系统、用户界面的发展,目前正朝着支持三维图形、支持多媒体及使用硬件芯片提高其性能等方面发展。

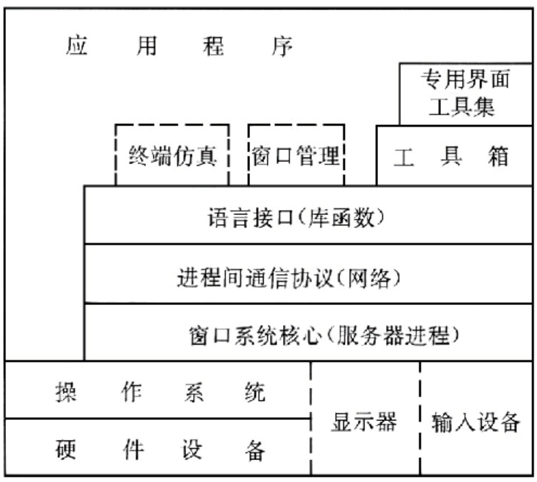


图 1 窗口系统的结构层次图

参考文献

1. 董士海. 窗口系统的内部结构及标准化. 计算机辅助设计及图形学学报,1990, 2(2): 35-40

2. Scheifler R W, Gettys J. The X window system. ACM Transaction on Graphics, 1986, 5(2): 79-109

(董士海)

cifa fenxi

词法分析(lexical analysis) 逐个读入源程序字符并按照构词规则切分成一列单词,再转换成词标流的过程。单词是语言中具有独立意义的最小单位,包括保留字、标识符、运算符、标点符号和常量等。词标是单词的机内表示,其格式由实现系统规定。

词法分析是编译过程中的一个阶段,在语法分析前进行。可以作为单独的一遍,将源程序转换成词标流供下一遍使用。也可以和语法分析结合在一起作为一遍,由语法分析程序调用词法分析程序来获得当前词标供语法分析程序使用。

在词法分析程序的设计和实现中,首先需要描述和刻画语言中的原子单位——单词,其次需要识别单词和执行某些相关的动作。描述程序设计语言的词法的机制是**3 型文法**和**正则表达式**,识别机制是有穷状态自动机。

在词法分析过程中,与语法分析无关的单词应预先处理。如为增加易读性而在保留字中引进的任选字,与语法语义分析无关,处理时可掠过,无须产生相应的词标。

参考文献

Aho A V, Sethi R, Ullman J D. Compilers principles, techniques and tools. Addison-Wesley, 1986

(钱树人)

ci cunchuqi

磁存储器(magnetic storage) 以硬磁材料为存储介质的计算机存储设备。在外磁场中的硬磁材料会被磁化,当外磁场消失后,硬磁材料仍保有磁性,其磁化方向与外磁场方向相同,因此可以用作存储介质。

硬磁材料保有的磁性只有两个方向,非常适合用来存储二进制的数据。做成的存储器有两种结构,一是固定式,存储介质与外加磁场间没有相对运动。如磁心存储器、磁杆存储器、磁膜存储器和磁

RAM。另一是运动式，记录和读出时存储介质与外加磁场间有相对运动。如磁鼓存储器、**磁带存储器**、**磁盘存储器**和磁光存储(MO)等。

磁心存储器 用磁心作为存储元件的存储器。磁心是用铁氧体材料经过压制、烧结而成的环状磁性物体。磁心的尺寸大约是0.8 mm×0.6 mm×0.2 mm(外径×内径×高)。每一个磁心可存储一位二进制数，根据存储器容量要求，把许多个磁心组成一个矩阵，例如32×32的矩阵，一个磁心相当于一个字中的一位。再把许多块板叠成一个立方体，板数等于字长的位数加上检验的位数，这样的立方体称为磁心体。它的存储量就是1024个字。磁心体和外围电路(译码电路、驱动电路、读出放大电路和控制电路等)组成一个磁心存储器。

第一个采用铁氧体材料的磁心存储器在1953年由美国麻省理工学院研制成功。由于它在性能、成本、体积各方面都优于当时的其他存储器，很快成为内存储器的主流产品。磁心本身可自动化生产，但穿线仍要人工进行，成本高。磁心本身不可能做得非常小，存取速度也慢，体积也大。因此，在半导体存储器研制成功后，磁心存储器很快就被淘汰了。

磁杆存储器 利用杆状磁性材料作为存储介质的半固定或固定存储器。其结构是在平板上布有许多绕在一起的初级绕组和次级绕组，每个绕组存储一位。磁杆用软磁材料制成。当绕组中间插有磁杆时，初级绕组和次级绕组间有耦合，电感大，表示这一位是1；绕组中没有磁杆，表示这一位是0。

磁杆存储器是一种**只读存储器**。随着半导体只读存储器(ROM)的研制成功并大量生产，磁杆存储器因在结构、体积、性能、成本等方面都不能与之相比而被淘汰。

磁膜存储器 利用薄膜磁性材料作存储介质，但是没有运动的存储器。磁膜存储器采用类似于磁心存储器那样的电流重合法进行选址、写入和读出，作为存储介质的材料是铁镍合金，用电镀、蒸发、溅射、沉积等工艺在玻璃或其他材料上制成薄膜，膜厚80～90 nm，面积约1 mm^2。由于磁膜是平面结构，可以用印制电路工艺来制作位线、字线和读出线，曾被认为可以代替磁心存储器。但实际上由于磁膜信号小，干扰大，器件体积大，工艺困难等原因，并没有得到广泛应用。

得到广泛应用的是记录和读出时存储介质与外加磁场间有相对运动的运动式磁存储器。在19世纪中叶，钢丝录音机就已发明。后来用磁带代替钢丝作存储材料的磁带录音机出现，并得到广泛应用。20世纪40—50年代研制的早期的计算机也曾采用磁带录音机作存储设备。

运动式磁存储器的存储介质是在带状、圆柱状或圆片状的基体的表面上附着的薄薄的一层磁性材料，因此也称为**磁表面存储器**。它仍是以磁性材料的两种不同剩磁状态或不同剩磁方向来表示二进制数字信息。

磁表面存储器的信息记录和读出过程都是电磁转换的过程，是通过磁头和运动着的磁介质来实现的。

磁头用软磁材料做铁心，上面绕有读写线圈。磁头铁心上有一条很窄的前间隙。工作时，磁头的前间隙靠近磁介质表面，附着有磁介质的基体相对于磁头做匀速运动。写入过程是将数字代码以脉冲电流的形式输入到磁头线圈中，形成磁场磁化磁介质，并以磁化状态的形式保存在磁介质上。读出过程则是通过磁头将磁介质上的磁化状态转换成电信号，再还原成数字形式。早期的磁头利用电磁感应原理读出信息，这种方式的灵敏度无法适应高密度记录。随着巨磁阻效应的发现，现代硬盘的磁头采用了巨磁阻效应磁头来读取信息，大大提高了信息检测的灵敏度，从而使记录密度大幅度提高。

磁表面存储器有磁鼓存储器、磁带存储器、硬磁盘存储器、软磁盘存储器和磁光存储器等。

磁鼓存储器 利用高速旋转的圆柱体的磁性表面作存储介质的存储设备。磁鼓存储器在20世纪50—60年代是计算机的主要外存储器。随着磁盘存储器的出现与发展，磁鼓存储器就被淘汰了。

磁带存储器 利用稳速运动的带状物体的磁性表面作存储介质的存储设备。磁带性能稳定，能长期使用和保存，而且价格低廉。磁带可以更换，可以脱机存储，因此适用于数据交换、文件档案的复制和保存以及数据的后备存储，还可用于数据输入。

硬磁盘存储器 利用高速旋转的盘状物体的磁性表面作存储介质的存储设备。从1956年第一台硬磁盘存储器IBM 350问世以来，由于它的存储容量大、数据传输速率高，一直是外存储器的核心设备。随着技术的不断进步，它的存储容量、存取速度和可靠性都有显著的提高，品种也非常丰富。

软磁盘存储器 利用高速旋转的薄片状物体的磁性表面作记录介质的存储设备。1972年研制成功单面单密度的200 mm(8 in)软磁盘机，随后又制成了双面双倍密度的130 mm(5.25 in)软磁盘机，

1981 年日本研制成 90 mm(3.5 in)软磁盘机。由于软磁盘的存储容量有限,软磁盘存储器的使用范围受到限制。由于软盘是可换的,曾广泛作为交换介质使用。由于采用闪存的 U 盘出现,软盘的这种交换功能被替代,现已退出市场。

磁光存储器 利用高速旋转的盘状物涂布的磁性表面作为粗存储介质的存储设备。写入时,利用激光快速加热记录介质表面,使其达到居里温度,以降低存储介质的矫顽力,并使用外部磁场改变存储介质的磁化方向;读出时,利用克尔效应,检测反射光的极化偏转角,检测出磁介质的磁化方向。磁光存储器在强调高可靠性、介质可更换的应用领域具有一定的优势。

磁表面存储器的技术进步是和数字磁记录的研究分不开的。提高水平磁记录密度的途径有:在介质方面,采用高矫顽力和高剩余磁化强度的磁介质,优化磁畴的微结构,使用 bit pattern 技术等;在磁头方面,减小磁头浮动高度,使用热磁辅助、微波辅助技术等。另外,利用磁盘驱动器强大的读出通道的检测能力,还可使用瓦记录技术。1976 年日本的岩崎俊一提出了垂直磁记录理论,其记录密度比传统的水平磁记录密度高 10 倍以上。磁表面存储器之所以能长期存在,是因为它具有许多优点:存储的内容不易丢失,可长期保存;存储密度不断提高,存储容量大,能满足计算机的发展需要;写入、读出的速度快,而且可以覆盖写。磁表面存储器也有其缺点:有机械运动部件,易磨损,可靠性差,寿命有限;功耗大;体积大。随着新型磁性材料的出现,磁头材料和结构的改进,新的记录方法的采用,在今后若干年内,磁记录的存储密度仍会有成倍的增长,其单位容量的成本优势还没有其他介质可替代。

近年来,自动盒带库、**磁盘阵列**这一类集成式磁存储器也在迅速发展,磁盘阵列已得到广泛应用。

(林兼 王海卫)

cidai cunchuqi

磁带存储器(magnetic tape storage)

一种以磁带作为存储介质,利用**数字磁记录**的原理,通过电磁变换传感器件(通常称为磁头)在稳速传动的磁带表面进行数据记录的顺序存取存储设备。它由磁带和**磁带机**(或称磁带驱动器)两部分组成。磁带作为存储信息的载体,是一种表面涂有一层薄薄的磁性材料的细长金属或者塑料带;磁带通常缠绕在可以转动的轴上,并用盒子进行保护;根据盒子类型的不同,分为开盘式磁带和盒带式磁带。

磁带存储器作为计算机的一种辅助存储器已有悠久的历史,在 20 世纪 40 年代后期最早问世的几台计算机中就已使用磁带存储设备,其技术源于当时的录音设备。经过几十年几代技术的改进,磁带存储器的位密度从 100 bpi 增至 370 kbpi,面密度从 1 kb/in^2 增至 1406 Mb/in^2,带速从 1.9 m/s 增至 6.3 m/s,数据传输速率从 7.5 kb/s 增至 250 MB/s,容量从 2.3 MB 增至 5 TB。

由于磁带具有体积小、容量大、成本低、寿命长、便携性等优点以及只能顺序存取等缺点,因此磁带存储器主要用于数据备份。磁带备份技术已经有 50 多年的历史,实践证明是一项非常成熟的技术。

(刘锡刚 胡迪青)

cidaiji

磁带机(magnetic tape drive)

一种能够在磁带上存取数据的装置(或称磁带驱动器)。磁带机的基本构成部分为:①读写磁头组件及读写电路;②使磁带快速启停并稳速行走的主动轮、驱动电机及伺服电路;③带盘电机及伺服电路;④磁带的缓冲机构;⑤防止磁带行走时左右晃动的导轮或导柱等导向机构;⑥数据写入和读出的计算机接口。当磁带机带动磁带移动时,通过磁头(电磁变换的传感器件)就可以对磁带上的数据进行读/写或者是擦除。

世界上第一台磁带机是 1952 年 IBM 公司推出的 IBM726。它利用真空缓冲带箱,使相对脆弱的磁带能够承受住系统的快速启动与停止,不会出现卡带现象,从此开创了磁带机的新局面,并奠定了一直沿用至今的半英寸带宽的通用标准。1984 年 IBM3480 盒带式磁带机问世,记录块间快速启停的开盘式磁带机逐步被新型流式连续传动的盒带式磁带机取代。目前,磁带机主要的规格形式有 DAT、8 mm、DLT、LTO、AIT 及 VXA 等。DAT(digital audio tape)最初是由惠普公司(HP)与索尼公司(SONY)共同开发,采用螺旋扫描磁记录技术;8 mm 技术由 Exabyte 公司在 1987 年开发,也是采用螺旋扫描技术;DLT(digital linear tape)由 DEC 和 Quantum 公司联合开发,技术源于 1/2 in 磁带机,主要定位于中、高级的服务器市场;LTO(linear tape open)技术由 HP、IBM、Seagate 于 1997 年联合制定,该技术有两种存储格式,即高速开放磁带格式 Ultrium 和快速访问开放磁带格式 Accelis。Ultrium 适用于备份、存储和

归档应用，Accelis 则主要用于处理在线数据及数据恢复；AIT(advanced intelligent tape)技术由 SONY 开发，采用螺旋扫描方式进行记录，在数据保护方面性能比较突出，主要用于数据备份；VXA 技术同样是由 Exabyte 公司开发，在保持高可靠性的基础上，提高了速度和容量。

根据装带方式的不同，磁带机可分为手动装带磁带机和自动装带磁带机，即自动加载磁带机。自动加载磁带机实际上是将磁带和磁带机有机组合起来，在单个磁带机中有一个更换装置，它能够从装有多盘磁带的磁带匣中取出磁带并放入驱动器中，或者执行相反的过程。（胡迪青）

cidaiku

磁带库(magnetic tape library) 一种具有大容量库体并自动加载磁带的大容量存储备份设备。它通常由多个**磁带机**子系统、1 台库管理计算机、1 个或者多个存储磁带的库体和 1 个盒式磁带装载机器人(或者是机械手)构成。磁带库备份数据时，机器人根据需要从库体中找到并抓取磁带，将其送入磁带机，然后把数据写入磁带，最后再经过机器人把磁带送回库体，如此反复实现数据保存，需要数据时，也可以找出磁带，读出数据。

世界第一台磁带库是 1974 年 IBM 公司推出的 IBM 3850。目前磁带库的容量可以从几十 TB 到几百 PB，远远高于**磁盘阵列**的容量。通常入门级的磁带库价格(2012 年)在 1 万美元左右，高端的从 20 万美元起，甚至可以达到上百万美元，但其单位容量费用却只有 10 美分/GB，不到其他存储器的 40%。磁带库的典型应用是数据备份和归档。另外，磁带库还可以在多级存储的海量数据系统中作为最底层的离线存储，用于存放较少使用的大量数据和文件。以 Oracle 公司的 StorageTek SL 8500 磁带库为例，该磁带库由用户接口模块、存储扩展模块、机械手扩展模块、驱动器及电子装置模块四种基本模块构成。可以支持 1448 ~ 10 000 个槽位，装配 4 ~ 64 个磁带机，目前容量可以达到 1000 PB。通过磁带库控制管理软件，还可将多个 SL 8500 连接起来，使槽位数超过 300 000 个。

管理软件、用于标记磁带的条码标签以及磁带自动装载机器人三者的应用，使得磁带库能够实现连续备份、自动搜索磁带，也可以在管理软件控制下实现智能恢复、实时监控和统计，整个数据备份过程完全自动化完成，不需要人工干预。

参考文献

Richard H D. Magnetic tape for data storage: An enduring technology. Proceedings of the IEEE, 2008, 96(11): 1775-1785

（胡迪青）

ciguangdie qudongqi

磁光碟驱动器(magneto-optical disc drive)

一种利用磁光效应的数字可重写光碟驱动器，是继**磁带机**、磁盘机之后，在 20 世纪 80 年代中后期发展起来的一种计算机外存储器，也叫磁光(MO)碟机。它具有存储容量大，存储媒体可换以及可顺序存取也可随机存取等特点。主要用于信息的海量存储，可作为计算机的配套外设，尤其适用于多媒体计算机系统中。由于磁光碟机的写、读是通过磁和光的共同作用来完成的，因此在名称上有“磁光”字样。由于记录媒体采用“居里点”高于常温的硬磁材料，在常温下其磁化状态不易被外界偏置磁场所改变，因此，磁光碟存储的信息在常温下非常稳定，数据保存时间长。

信息写入时，当聚焦激光的能量使记录媒体表面某个极小区域温度上升到存储材料“居里点”以上时，该区域内的媒体就会容易地随外界偏置磁场的变化而迅速被磁化并达到饱和，这样就可像磁盘一样进行写入或抹除信息。信息读出时，则是利用磁场对线性偏振光的克尔(Kerr)效应来实现的。当线性偏振光入射到记录媒体上并反射时，会根据媒体上表征记录信息的不同的磁化方向，使反射光的偏振方向发生改变，这种现象就是所谓的克尔效应(Kerr effect)。两种不同磁化方向所对应的反射偏振光的偏振方向之间的夹角称为克尔角。在磁光碟机中，信息的读出实际上就是检测克尔角。

依据上述基本原理实现信息写入和读出的磁光碟驱动器，一般包括下述主要部分：①带动磁光碟旋转的主轴电机；②完成用激光向磁光碟上写入和从光碟上读出信息的光学头；③配合信息的抹除和写入、用来产生不同方向偏置磁场的写抹电磁铁(磁头)；④带动光学头寻找记录道的直线电机，把光碟头定位在记录道上并保证激光束聚焦在光碟表面的二维执行机构；⑤控制磁光碟机工作的电子线路板等。

光碟直径尺寸主要有 130 mm(5. 25 in)和 90 mm(3. 5 in)两种。其中，130 mm 的磁光碟使用双面记录，其容量范围可从 650 MB 到 9. 1 GB；130 mm 磁光碟驱动器主要使用 SCSI 接口。90 mm

磁光碟只采用单面记录，其容量范围为 128 MB 至 2.3 GB；90 mm 磁光碟驱动器采用 SCSI、IDE、USB 等多种接口形式（参见**外存储设备接口**）。

衡量磁光碟机性能和影响其使用的主要技术指标有：容量、数据传输速率、平均存取时间、旋转速度、接口、外形尺寸和平均无故障时间等。

磁光碟与磁光碟驱动器的主要市场在日本，其主要研究机构和生产商也在日本。磁光碟是一种可换的介质，保存时间长，适于多媒体数据记录和医学档案记录，在广告行业有较广泛的应用。

参考文献

杨建东，裴先登．可从写光盘机与多功能光盘机技术．电子计算机外部设备，1993，(3)：38-42

（高宝石　王海卫）

cipan cunchuqi

磁盘存储器（magnetic disk storage）　一种以磁性圆盘为存储介质，盘片按一定转速旋转，驱动载有磁头的磁头臂到达且稳定在指定的半径位置上，控制磁头在盘面磁层上按一定的记录格式和编码方式进行写入和读出的存储设备。磁盘存储器通常由磁盘、磁盘驱动器（或称磁盘机）构成。磁盘按基片材料分为硬磁盘与软磁盘两类。硬磁盘的盘基用非磁性轻金属材料或玻璃制成，软磁盘的盘基用挠性塑料制成。根据盘片的安装方式，磁盘驱动器又可分为可换磁盘驱动器与固定磁盘驱动器两类。固定硬磁盘驱动器是广泛使用的类型，软磁盘驱动器则只有可换磁盘一种类型。

不论是**硬磁盘驱动器**还是**软磁盘驱动器**，作为驱动器的结构和工作原理是大同小异的。磁盘驱动器的主轴带动盘片按一定转速稳定旋转，并按照主机发来的命令进行磁道的寻找（头臂定位），磁头的选接，数据的写入（存）与读出（取）等操作。

与磁盘存储器有关的参数和性能指标如下。

存储面数　指可存储数据的盘片的表面数。对于软磁盘驱动器，存储面数为 2。对于硬磁盘驱动器，存储面数是盘片数的 2 倍。例如，2 片的硬磁盘，存储面数为 4 面。一般每面只用一个磁头，存储面号也就是磁头号。

磁道　磁盘存储表面被分成许多同心圆，每一个同心圆就是一个磁道。磁道有一定宽度，是存储信息的区域。

扇区　每个磁道所在的同心圆分成若干等分的段，每一段称为一个扇区，每个扇区存储的字节数是一定的。软磁盘驱动器的各个磁道的扇区数是相等的。硬磁盘驱动器在等位密度记录的情况下，一个盘面各个磁道的扇区数是不相等的，盘面内侧直径小的磁道扇区数少，盘面外侧直径大的磁道扇区数多。

道柱面　在硬磁盘驱动器中，把所有的盘面上的同一直径的磁道看成一个整体，就像一个圆柱体的外表面一样，称为道柱面，简称柱面。

存储密度　通常用道密度，位密度和面密度来评定。

（1）道密度　指沿磁盘的盘径方向单位长度内的磁道数目。道密度习惯上用每英寸的磁道数来表示，记作 tpi；或用每毫米的磁道数来表示。

（2）位密度　指沿磁道圆周方向单位长度内所存储的二进制数的个数。习惯上用每英寸存储多少位来表示，记作 bpi；或用每毫米存储多少位来表示。

（3）面密度　指单位面积上存储的二进制数的个数，它等于道密度与位密度两者之乘积。

存储容量　磁存储器所能存储的二进制数的总量，以位数或字节数计量。硬盘的容量以兆字节（MB）或千兆字节（GB）为单位，1 GB = 1024 MB。但硬盘厂商在标称硬盘容量时通常取 1 GB = 1000 MB，1 TB = 1000 GB。

存储容量有未格式化和格式化两种。未格式化容量指的是，在磁盘机的全部存储面上所有磁道所能存储的二进制码的总量。所谓格式化，就是指计算机系统按照要求在数据存入之前在磁道上先写入地址、识别码和同步码等与地址有关的信息，目的在于能迅速辨认所存取的数据所在位置。格式化容量按下式计算

格式化容量 =（存储字节数每扇区）× 扇区总数

对于各个磁道的扇区数是相等的磁盘，其格式化容量可按下式计算

格式化容量 =（存储字节数每扇区）×（扇区数每磁道）×（磁道数每面）× 面数

显然，格式化后，与地址有关的信息要占去一定的存储容量，格式化容量低于未格式化容量。

存取时间　指主机发出指令到数据传输结束所需的时间，即磁头从当前所在位置移动到目标位置（目标磁道）并稳定下来，然后从目标磁道上寻找到要读写的扇区并进行读写所需的全部时间，它由寻道时间、等待时间和传输时间三部分组成。

（1）寻道时间　磁头从所在位置运动到目标磁道所需的时间。显然，运动距离最小即寻找相邻磁

道的寻道时间最短,而运动距离最大即从首道寻找到末道(或相反)的寻道时间最长。前者称为最小寻道时间,后者称为最大寻道时间。平均寻道时间是采用统计平均值求得的,一般等于运动距离为最大距离的1/3时所需的时间。

(2) 等待时间　指磁头到达目标磁道后等待要读写的扇区旋转到达磁头下方所需的时间。最短是0,最大是磁盘旋转一周所需的时间。显然,平均等待时间是磁盘旋转一周所需时间的一半。

(3) 传输时间　数据从内存写到盘面或从盘面读出送到内存所需的时间。传输时间在存取时间中所占比例很小。

数据传输速率　在单位时间内磁盘存储器传送的二进制数的位数或字节数。***数据传输速率***的单位为Mb/s或MB/s。

硬盘存储器的数据传输速率有外部数据传输速率和内部数据传输速率之分。外部数据传输速率指硬磁盘驱动器和主机之间的数据传输速度,它与所采用的接口有关;内部数据传输速率是盘面与高速缓存之间的数据传输速度,它等于位密度与磁道线速度的乘积。

磁盘驱动器的存储密度和存储容量还将会有大幅度增长,其单位容量的成本是目前所有存储介质中最低的。磁盘存储器将继续在计算机辅助存储设备中占重要地位。

参考文献

张江陵,金海. 信息存储技术原理. 武汉:华中科技大学出版社,2000　(刘锡刚　周功业)

cipan zhenlie

磁盘阵列(magnetic disk array)　用多台**磁盘存储器**按数据分块与冗余信息容错,以矩阵形式组成的快速大容量外存储子系统。它在阵列控制器的组织管理下,能实现数据的并行、交叉存取存储操作。由于阵列中的一部分容量存放有冗余信息,一旦系统中某一磁盘失效或存取通道失效,利用冗余信息可以重建用户数据。磁盘阵列是一个包括许多品种的泛称,常见的一种称为廉价冗余磁盘阵列(redundant arrays of inexpensive disks,RAID),进一步发展成了独立(independent)冗余磁盘阵列(RAID)。构造磁盘阵列的思路适用于构造其他存储器阵列,如**固态盘**阵列、只读光盘(CD-ROM)阵列和**磁带存储器**阵列,并可用于分布式网络存储系统上。

磁盘阵列是在**中央处理器**性能逐年增强,而输入输出速度受限,存储容量又与日俱增的背景下产生的。磁盘阵列的性能在许多方面超过单台大型存储设备的性能,而价格则低于同容量的单台大型设备,特别是采用冗余纠错技术后,其可用性大为增强,因而得到很大发展。

磁盘阵列的构思渊源于早期提出的拆分、交叉存取、分块等概念。拆分的意思是将数据分割为小条,按条并行存取;交叉存取是指在多台磁盘存储器上进行交叉、并行操作;而分块的含义则是对群集的数据分成较大的块,将大块数据分布到若干台磁盘存储器上。它们的含义有相似之处,但存在着某种差别。拆分是针对主机请求读、写数据而言的,交叉存取是从减少存取时间的意义上说的,而分块则是针对存储空间的有效利用而采取的。三者分别用于处理三个层面上的并行性问题。

磁盘阵列在构成形式上分三类,一是外接式磁盘阵列柜、二是内接式磁盘阵列卡,三是利用软件来仿真。

外接式磁盘阵列柜最常用于大型服务器上,具可热插拔(Hot Swap)的特性,不过这类产品的价格都很贵。

内接式磁盘阵列卡,价格较低,但需要较高的安装技术,适合技术人员使用操作。

利用软件仿真的方式,由于会消耗大量机器时间,不适合大数据流量的应用。

1987年,美国D. A. Patterson等人把廉价冗余磁盘阵列(RAID)分成5级,即RAID 1~RAID 5。后来人们又增加了RAID 0、RAID 6和RAID 7共为8级。

(1) 0级冗余磁盘阵列(RAID 0)　一种不具备容错能力的阵列。连续以位或字节为单位分割数据,并行读/写于多个磁盘上,因此具有很高的数据传输率,但它没有数据冗余,不能算是真正的RAID结构。

(2) 1级冗余磁盘阵列(RAID 1)　采用镜像容错改善可靠性的一种磁盘阵列。阵列中磁盘分成若干组,每一组由两个相同的盘组成。每次写数据时同样地写在一组中的两个盘上,因此其可靠性很高。它的平均无故障时间(MTTF)远大于单台磁盘存储器,而且在某一磁盘存储器失效后,经更换新的之后,数据可以重构。但是磁盘阵列的有效容量减少到只有总容量的一半。

10/01级冗余磁盘阵列(RAID 10/01)

将RAID 0和RAID 1标准结合的产物,在连续

地以位或字节为单位分割数据并且并行读/写多个磁盘的同时,为每一块磁盘作磁盘镜像进行冗余。

(3) 2 级冗余磁盘阵列(RAID 2)　采用汉明码作错误检验的一种磁盘阵列。

将数据条块化地分布于不同的硬盘上,条块单位为位或字节,并使用汉明码的编码技术来提供错误的检测及纠错。这种编码技术需要多个磁盘存放检查及纠错信息,使得 RAID 2 技术实施复杂。

(4) 3 级冗余磁盘阵列(RAID 3)　采用奇偶校验码的并行传送一种磁盘阵列。这种校验码与 RAID 2 不同,只能查错不能纠错。鉴于磁盘存储器本身多数已采用了循环冗余检验码检测错误,只要在阵列中能够检测出错误出现在哪台磁盘存储器上,便能达到纠正错误的目的,检验盘数目可减少到只有一个,它可以识别出一台出错的磁盘存储器。

(5) 4 级冗余磁盘阵列(RAID 4)　采用带奇偶校验码的独立磁盘结构的磁盘阵列,它也只用一个检验盘。RAID 4 和 RAID 3 很像,不同的是,RAID 3 按位或字节交叉存取,而 RAID 4 则是按扇区交叉存取。因而对于小量传输数据可以单独地对某个盘操作,无须像 RAID 3 那样完成一次小量数据写入要涉及全组,RAID 4 只牵涉组中的两台磁盘存储器(一台数据盘、一台检验盘),因而提高了小量数据的输入输出速率。不过在失败恢复时,它的难度可要比 RAID 3 大得多。

(6) 5 级冗余磁盘阵列(RAID 5)　采用分布式奇偶校验信息块独立磁盘结构的磁盘阵列。它不设置固定的专作检验用的磁盘存储器,而按某种规则在组内临时指定检验盘。于是在同一台盘上既记录有数据,也记录有检验信息。这一改动,解决了争用检验盘的问题。与 RAID 3 相比,RAID 3 每进行一次数据传输,需涉及所有的阵列盘。而 RAID 5 大部分数据传输只对一块磁盘操作。

(7) 6 级冗余磁盘阵列(RAID 6)　带有两个分布奇偶校验信息块独立磁盘结构的磁盘阵列。两个独立的奇偶系统使用不同的算法,数据的可靠性非常高,即使两块磁盘同时失效也不会影响数据的使用。较差的性能和复杂的实施方式使得 RAID 6 很少得到实际应用。但是,在某些安全性要求很高的环境下,RAID 6 是一种较好的解决方案。

(8) 7 级冗余磁盘阵列(RAID 7)　一种新的 RAID 标准,其自身带有智能化实时操作系统和用于存储管理的软件工具,可完全独立于主机运行,不占用主机 CPU 资源。RAID 7 可以看作是一种存储计算机(Storage Computer),它与其他 RAID 标准有明显区别。

阵列控制器　一般由**微处理器**、**高速缓冲存储器**、接口、适配器等组成。阵列控制器的主要功能是:①完成命令管理,包括接收主机命令,解释和分解命令,形成控制信号,回收磁盘存储器的工作状态信号以及协调盘阵列各部分的工作等;②进行地址变换,包括数据分割(拆分和分块)和聚合,地址变换,坏块管理;③实现数据的检错、纠错和重构以及最优化调度。为了扩展阵列的功能,还可以在盘阵列控制器中设置某些功能部件,如数据的过滤与排序、数据的压缩与解压等。

此外,不使用单独的控制器,仅以软件实现上述功能的磁盘阵列称为**软磁盘阵列**。

使用固态盘(SSD)根据上述规则组成的阵列称为**固态盘阵列**(SSD-RAID)。SSD 有着性能高、体积小、发热低的优点,组成的阵列一般读写性能强大、便于携带、发热较小。但是由于目前 SSD 成本高、容量小使得 SSD-RAID 实际应用较少,一般用于高性能服务器的存储系统。

分布式磁盘阵列　建立在多台计算机上的具有单一地址空间的嵌入式磁盘子系统。分布式磁盘阵列的计算机之间一般使用高速的网络连接,使应用程序在访问速度上意识不到分布式结构的存在。与传统集中式磁盘阵列相比,分布式磁盘阵列有以下几个好处:①提供更高的可靠性,特别是在单个计算机体统出现故障时。可以通过其他计算机系统上的冗余信息来防止数据丢失。通过把计算机分布在不同的地点,可以实现灾难恢复。②支持多计算机之间的并行 I/O 处理,增加带宽。可以实现系统中计算机之间的负载平衡。③很容易在分布式 RAID 的基础上实现附加功能。比如:快速的备份和恢复;当增加磁盘或者新的服务器时,由自动配置软件透明实现系统的容量增加和 I/O 处理能力提高。

集中式 RAID 一般是作为存储子系统连接到存储服务器上,或者作为网络附属存储磁盘直接连接到网络上。分布式磁盘阵列的多个联网计算机的存储系统,由软件或中间件连接起来实现 RAID 的功能,不存在一个集中的存储服务器,是一种分布式结构体系。

磁盘阵列除应用于大型计算机外,还广泛应用于小型机和服务器,在现代数据中心中,磁盘阵列是主流的海量存储设备。

参考文献

1. 张江陵,冯丹. 海量信息存储. 北京:科学出版社,2003

2. 曹强,黄健忠,万继光,谢长生. 海量网络存储系统原理与设计. 武汉:华中科技大学出版社,2010

(张江陵 万继光)

cong yundong huifu jiegou

从运动恢复结构(structure from motion, SFM) 从二维的图像序列中恢复出场景的三维几何结构和摄像机运动的过程。当摄像机与某一个刚体间发生相对运动时,给定刚体上的一组点在拍摄的图像序列中的二维运动轨迹,SFM 的目标是恢复出这组点在固定的世界坐标系中的三维坐标和每个时刻摄像机相对于世界坐标系的运动参数。假定摄像机的正交投影模型,m 个点的 n 帧二维轨迹可提供 $2mn$ 个方程(每点每帧提供关于图像二维坐标的 2 个方程),而共有 $(3m+6n)$ 个未知数(每个点的三维坐标和每个时刻摄像机相对于世界坐标系的旋转和平移)。当 m 和 n 足够大时,$2mn>3m+6n$,这样就有足够的信息可以把点的三维坐标和摄像机运动参数求解出来。求解过程可以通过矩阵的奇异值分解和利用旋转矩阵的正交性来完成。SFM 可以用于摄影测量、从图像序列中自动重构虚拟世界模型、得到摄像机的运动参数用于驾驶辅助系统、增强现实等。

参考文献

1. Tomasi C, Kanade T. Shape and motion from image streams under orthography: a factorization method. International Journal of Computer Vision, 1992, 9(2):137-154

2. Forsyth D A, Ponce J. 计算机视觉:一种现代方法. 林学訚, 王宏,等译. 北京:电子工业出版社, 2004

(邱慧军)

cucaoji lilun

粗糙集理论(rough set theory) 一种能定量处理不精确、不一致、不完整信息与知识的扩展集合理论。该理论通过定义两个确定的集合(上近似集、下近似集)来近似表达一个具有不确定边界的集合概念,是一种采用精确集合概念来描述不精确集合概念的不确定集合理论。由于上近似集和下近似集是可直接从数据中计算获取的精确集合,粗糙集理论摆脱了对一些先验知识的依赖,如先验概率、隶属度函数形式等。

该理论由波兰学者 Zdzislaw Pawlak 于 1982 年提出,是**不确定性推理**、**机器学习**、**数据挖掘**和决策支持等的一个重要理论方法,可通过**关系数据库**分类归纳形成概念和规则,通过等价关系的分类及其对目标的近似,实现知识发现。

信息表知识表达系统 一个信息表知识表达系统 S 可以表示为 $S=\langle U,R,V,f\rangle$,其中,U 是对象全集,也称为论域,$R=C\cup D$ 是属性全集,子集 C 和 D 分别称为条件属性集和结果属性集,$V=\cup_{r\in R}V_r$ 是属性值的集合,V_r 表示属性 $r\in R$ 的属性值范围,即属性 r 的值域,f: $U\times R\rightarrow V$ 是一个信息函数,它指定 U 中每一个对象 x 的属性值。

不分明关系 对于任一属性集合 $B\subseteq R$,定义一个不可分辨二元关系(不分明关系)$\mathrm{IND}(B)=\{(x,y)\mid(x,y)\in U^2,\forall_{b\in B}(b(x)=b(y))\}$。

上近似集与下近似集 给定信息表知识表达系统 $S=\langle U,R,V,f\rangle$,对于任一对象集合 $X\subseteq U$ 和属性集合 $B\subseteq R$,X 关于 B 的上近似集($B^-(X)$)和下近似集($B_-(X)$)分别定义如下:

$$B^-(X)=\cup\{Y_i\mid(Y_i\in U\mid\mathrm{IND}(B)\wedge Y_i\cap X\neq\varnothing)\},$$

$$B_-(X)=\cup\{Y_i\mid(Y_i\in U\mid\mathrm{IND}(B)\wedge Y_i\subseteq X)\},$$

其中,$U\mid\mathrm{IND}(B)=\{X\mid(X\subseteq U\wedge\forall_{x\in X,y\in X,b\in B}(b(x)=b(y)))\}$ 是不分明关系 B 在 U 上的划分。

上近似集和下近似集也可以通过集合形式作如下定义:

$$B^-(X)=\{x\mid(x\in U\wedge[x]_B\cap X\neq\varnothing)\},$$

$$B_-(X)=\{x\mid(x\in U\wedge[x]_B\subseteq X)\},$$

其中,$[x]_B\in U\mid\mathrm{IND}(B)\wedge x\in[x]_B$,即 $[x]_B$ 是对象 x 所在的划分块。

集合 $\mathrm{BN}_B(X)=B^-(X)-B_-(X)$ 称为对象集合 X 关于属性集合 B 的边界域;$\mathrm{POS}_B(X)=B_-(X)$ 称为对象集合 X 关于属性集合 B 的正域。

可定义集 给定信息表知识表达系统 $S=\langle U,R,V,f\rangle$,对于任一对象子集 $X\subseteq U$ 和属性子集 $B\subseteq R$,当且仅当 $B^-(X)=B_-(X)$,称集合 X 是 B 可定义集。

粗糙集(rough set) 给定信息表知识表达系统 $S=\langle U,R,V,f\rangle$,对于任一对象子集 $X\subseteq U$ 和属性子集 $B\subseteq R$,当且仅当 $B^-(X)\neq B_-(X)$,称集合 X 是 B 粗糙集。

粗糙度(roughness) 给定信息表知识表达系统

$S=\langle U,R,V,f\rangle$,对于任一对象子集 $X\subseteq U$ 和属性子集 $B\subseteq R$,定义 X 关于 B 的粗糙度为:

$$P_B(X)=1-\frac{|B_-(X)|}{|B^-(X)|},$$

其中,$X\neq\varnothing$(如果 $X=\varnothing$,可定义 $P_B(X)=0$);$|X|$ 表示集合 X 中包含元素的个数。

弱对象集合 X 是论域 U 上的一个关于属性集合 B 的粗糙集,则:

(1) 如果 $B_-(X)\neq\varnothing\wedge B^-(X)\neq U$,则称 X 为 B 粗可定义的;

(2) 如果 $B_-(X)=\varnothing\wedge B^-(X)\neq U$,则称 X 为 B 内不可定义的;

(3) 如果 $B_-(X)\neq\varnothing\wedge B^-(X)=U$,则称 X 为 B 外不可定义的;

(4) 如果 $B_-(X)=\varnothing\wedge B^-(X)=U$,则称 X 为 B 全不可定义的。

粗糙集运算

并运算　$B_-(X\cup Y)\supseteq B_-(X)\cup B_-(Y)$, $B^-(X\cup Y)=B^-(X)\cup B^-(Y)$;

交运算　$B_-(X\cap Y)=B_-(X)\cap B_-(Y)$, $B^-(X\cap Y)\subseteq B^-(X)\cap B^-(Y)$;

差运算　$B_-(X-Y)=B_-(X)-B^-(Y)$, $B^-(X-Y)\subseteq B^-(X)-B_-(Y)$;

补运算　$\sim B_-(X)=B^-(\sim X)$, $\sim B^-(X)=B_-(\sim X)$。

粗糙集理论已经在故障诊断、决策支持、图像处理、智能控制、信息安全、生物信息处理等领域取得一系列成功应用。目前,关于粗糙集的研究主要集中在扩展粗糙集理论模型、基于粗糙集的粒计算、粗糙集知识约简的时间效率与泛化能力、粗糙集与其他相关不确定性处理理论的融合等方面。

参考文献

1. Pawlak Z. Rough sets: Theoretical aspects of reasoning about data. Kluwer Academic Publishers, 1991

2. 王国胤. Rough 集理论与知识获取. 西安: 西安交通大学出版社. 2001

（王国胤）

cunchu anquan

存储安全(storage security)　保障存储系统安全运行的基本技术,用于维护存储系统中数据的机密性、完整性和可用性。存储安全作为信息安全的"最后一道屏障",越来越受到业界的重视。

网络存储工业协会(Storage Network Industrial Association, SNIA)将存储安全表述为确保数字资产安全所使用的存储、网络以及安全方面的准则、技术和方法的集合。存储安全涉及网络安全性和存储自身安全性,存储安全既要保证信息在不可信网络中的传输安全,又要保护数据在存储设备上的存取安全。

存储安全可以借鉴网络安全领域中已有技术,如身份验证、数据加密、数字签名等,保证数据在传送和存储中的机密性和完整性,还要研究不同于网络安全中的安全协议和机制以满足存储安全对访问控制、密钥管理、存储资源分配和管理的特殊需求。已有的存储安全技术主要是将传统信息安全的技术应用于存储系统,为某一特定应用提出专门的解决方案,例如增强文件服务器的安全性、客户端加密文件系统、存储系统的入侵检测等。从具体使用的技术来看,存储安全技术主要包括以下几个方面:

(1) 存储认证　存储认证是存储安全的第一道门户,被定义为存储系统实体身份或消息源的确认过程,分别对应存储系统实体认证和存储系统消息认证。在进行授权访问(如读或写)前,存储系统应该验证数据的提供者、使用者和管理员的身份。

(2) 存储访问控制　存储访问控制是存储系统提供的基本安全措施,决定了谁能够访问存储系统、能访问存储系统的何种资源以及如何使用这些资源。适当的存储访问控制能够阻止未经允许的用户有意或无意地获取存储系统的数据。存储访问控制的手段包括登录控制、权限授予与回收、权限核查等。

(3) 存储数据机密性和完整性　随着存储系统的发展,越来越多的存储设备暴露于网络环境下,攻击者可以修改网络传送或存储设备中的数据,也可以用有效的老版本数据代替当前版本的数据。存储数据机密性通常采用对数据加密的办法来解决,而存储数据完整性可以用数字签名和消息认证码来解决。

(4) 存储审计　存储审计是一种用户来验证数据被安全存放在存储系统的手段。用户在没有数据原始拷贝的情况下,通过交互式协议要求存储系统根据用户指定的随机值和原始数据来产生能证明数据存在的证据,用户则通过验证该证据的有效性来确保数据被安全存储。

(5) 存储入侵检测　存储入侵检测是指存储系统通过分析数据访问信息以检测是否受到非法入侵的技术手段。在存储系统不同层上配置存储入侵检

测并将不同层上的事件联系起来可以提高存储入侵检测的效率。

(6) 数据自毁　数据自毁是指数据拥有者或具有相关权限的共享者根据数据的安全需要定义一系列数据销毁触发条件(比如数据过期时间、数据访问次数、数据被非法访问等),存储系统根据这些触发条件对数据进行销毁操作(如删除数据加密密钥或对数据进行随机字节覆盖写等)。数据自毁过程不需要数据拥有者、共享者或系统管理员的参与,增强了数据防泄漏保护。

面对存储系统网络化和规模化所带来的种种安全挑战,存储安全解决方案在确保数据安全的同时,必须保证存储系统可扩展性、可管理性和高可用性,权衡存储安全的各个组件,在满足用户安全需求的同时将安全协议的开销降到最低。

参考文献

Hibbard E A, et al. Standards relevant to storage security. SNIA Internal Report Volume 9, Storage Networking Industry Association, 2009 (周可　雷栋梁)

cunchu baohu

存储保护(memory protection)　在多道程序和多处理机系统中,为使多个用户在共享**主存储器**时,能防止某一个用户程序出错而破坏在主存储器中其他用户程序或系统程序而设置的一种功能。

存储保护功能是靠计算机硬件和软件配合实现的。存储保护功能可有对计算机工作状态的保护、对存储区域的保护和对访问方式的保护。

1. 工作状态保护

工作状态保护是为防止某一个程序出错而影响整个计算机系统工作而设置的一种保护功能。大多数计算机在执行程序时把工作状态分为两种:一种是执行操作系统或系统管理程序时所处的状态,称为**特权状态**或**管态**;另一种是执行用户程序时所处的状态,称为**非特权状态**或**目态**。在计算机中规定:能改变机器运行状态和用户对机器中资源使用权限的指令称为特权指令,只有在操作系统和系统管理程序中才能使用,也就是说,只有在特权状态下才能执行特权指令。如果用户程序误用特权指令,就可能对整个计算机系统产生严重破坏,因此在非特权状态下使用特权指令时,计算机能进行检查,发出错误中断,进行保护。

2. 存储区域保护

存储区域保护是为防止某一个程序对同时存放在主存储器中的其他程序进行错误访问而设置的一种保护功能。存储区域保护主要有以下几种方法。

(1) 界限寄存器保护　界限寄存器保护在分区管理的主存储器中,对每个存储区域设置一个上界寄存器和一个下界寄存器,分别放置分配给一道程序的存储区域的上界地址和下界地址。在执行程序时,每当进行逻辑地址到物理地址的变换时,均需把变换所得的物理地址同这对上界寄存器和下界寄存器的内容进行比较,如果小于下界地址或大于上界地址,则说明这道程序越出自己的存储区域,这时计算机就要发出越界中断,进行保护。此外,也可以为每个存储区域设置一个基址寄存器和一个限长寄存器,用它们来检查是否越界,实现界限保护。

(2) 键保护　键保护通常用于页式或段页式存储器中。对主存储器的每个页面都设有几位叫作存储键的编码,对每道程序也分配位数与存储键相同的编码,叫作保护键。当一道程序要访问主存储器的某一个页面时,先要把这道程序的保护键同被访问页面的存储键相比较,只允许程序访问那些存储键与其保护键相同的页面,如果检查出不一致,就会发生错误中断,进行保护。存储键和保护键都是由操作系统来设置的。

(3) 段保护　界限寄存器保护和键保护都是对访问主存储器时的物理地址出错进行保护的,而段保护可以在逻辑地址变换成物理地址之前出错时,检查出错误,进行保护。在段式存储器或段页式存储器中,每个用户只能访问分配给自己的段。段式存储器中逻辑地址分成段号和段内地址两部分,如果段号超出段表总的项数,或段内地址超出这个段的段长(段表项中有段长栏),则说明有错误。同样,在段页式存储器中,逻辑地址分成段号、页号和页内地址3部分,如果段号超出段表总的项数,或页号超出这个段内总的页数(段表项中有页表长栏),也说明有错误。这时会发生错误中断,进行保护。

(4) 环保护　前面3种保护方法只能使某一个程序不去破坏其他程序,环保护可以保护正在执行的程序自己,这对于系统程序尤为重要。环保护是把程序按其对整个系统能否正常工作的影响程度分成几层,每层规定了访问权限,越里面的层,级别越高。一个层只能访问本层或外面级别较低的层,不能访问内部较高层次的程序,如图1所示。这样可以防止低层程序出错影响到高层的程序。在图1的例子中,把主存储器分为4层,从里到外为环0到环

3。把系统程序中最重要的核心部分放在环 0 中,比较不重要的部分依次放在环 1 和环 2 中,用户程序则在环 3 中(用户程序当然也可以按其重要性分设几个环)。在处理器中设有现行环号寄存器,放置正在处理器中执行的段的环号,它由操作系统设置。当某一个程序从 1 个段转移到别的段去的时候,或是要从别的段中读写数据时,先要进行检查,如果要进入的段的环号比现行环号小,就要发出错误中断,进行保护。

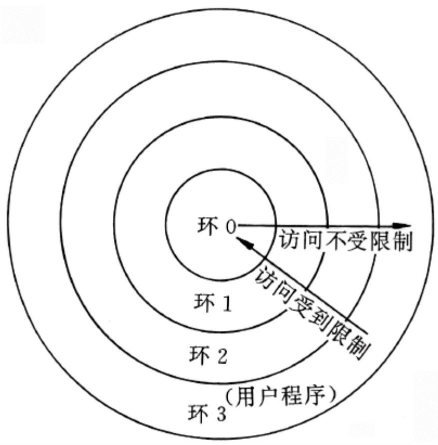


图 1　环保护

3. 访问方式保护

对存储区域的保护还可以按不同的访问方式加以保护,即把存储区域进一步区分为是数据区域还是程序区域,是只能读出的区域还是可以写入的区域。上述的几种存储区域保护方法都可以加上访问方式保护,其方法是:对于每个存储区域设置访问方式位(分读、写和执行 3 种)。读位控制这个存储区域是否允许读出,写位控制它是否允许写入,执行位控制从这个存储区域中读出的数据是否可作为指令来执行。例如,从执行位为 0 的区域中取出数据当指令执行,就说明有错。同样,如果向写位为 0 的区域写入数据,说明有错。凡检查出这类错误,会发出错误中断,以使程序在发生这些错误时不会造成严重后果。

参考文献

孙强南,孙昱东. 计算机系统结构. 北京:科学出版社,1992

(孙强南　张广艳)

cunchu guanli

存储管理(memory management)　为了有效地使用计算机系统中的存储资源而进行的管理。存储管理是由**操作系统**在有关硬件的配合下进行的。存储管理包括存储单元的分配和回收、程序和数据在**主存储器**和**辅助存储器**之间的交换、程序再定位、地址转换、**存储保护**等。

存储管理是随着**多道程序设计**和**分时处理**的出现而发展起来的。在计算机系统中,由于**中央处理器**工作速度快,而输入输出设备的工作速度慢,这一矛盾促成了多道程序和分时处理的出现和发展。要在同一台处理机上同时运行多道用户程序,需要把存储器划分成多个部分来存放多个程序,因而产生了有效地进行存储管理的需要。

为了保证中央处理器能连续不断地工作,要求主存储器中能存放的程序和数据越多越好。但主存储器的容量不可能无限扩大,因而出现了交换技术。交换是将一个(或一部分)程序从主存储器中取出来写到辅助存储器中,同时将另一个(或另一部分)程序从辅助存储器读到主存储器中去的操作,前一操作称为换出,后一操作称为换入。有了交换功能,就能通过操作系统的自动调度,把主存储器和辅助存储器统一成一个更大的存储器空间。

在存储管理中,需要把存储器划分成多个部分以使多个用户能共享主存储器。一种简单的办法是把主存储器划分成多个地址连续的存储区,给每个用户进程分配 1 个区。划分是以用户进程为单位的,这时交换也以用户进程为单位进行。

采用分区办法进行存储管理时,由于用户进程长短不等,随着不断进行换入换出,主存储器中会产生很多碎片,即夹在两个用户进程之间的零星的空闲存储单元。碎片一般不大,放不下 1 个用户进程,无法加以利用。为了去掉这些碎片,需要时时采取压紧操作,即通过操作系统移动主存储器中的有关进程,把夹在进程间的碎片集中在一起,从而可以加以利用。

交换和压紧操作都要改变进程在主存储器中的位置。当进程中的指令在存储器中的位置变动后,指令中的地址码(操作数地址和转移地址)也必须随之变动,这就产生了指令在主存储器中浮动的需要,从而有了把地址分为**逻辑地址**和**物理地址**的需要。物理地址又称为实际地址,它是存储单元在主存储器中的实际位置;逻辑地址则是指令或数据在程序中相对于程序开始点的相对位置。逻辑地址是在程序设计中使用的,它只有按一定规则经过变换后才能成为物理地址,只有用物理地址才能在主存储器中找到所需要的存储单元。把逻辑地址按一定

规则变换成物理地址的操作，称为**程序再定位**。

根据操作时间的不同，把程序再定位操作分为静态程序再定位和动态程序再定位两种。前者是在程序初始装入主存储器时，由定位装入程序进行的程序再定位；后者则是在程序执行过程中，在对指令或数据进行访问时，才由地址变换机构执行的程序再定位。静态程序再定位靠软件进行，动态程序再定位靠硬件进行。

在分区和交换技术基础上发展起来的存储管理有三种基本方式，即分页存储管理、分段存储管理和段页存储管理，它们采用的存储系统分别为页式存储系统、段式存储系统和段页式存储系统。这些存储系统中的逻辑地址和物理地址之间的转换可参见**虚拟存储器**。分页存储管理和段页存储管理已普遍应用于现代计算机中。

参考文献

孙强南，孙昱东. 计算机系统结构. 北京：科学出版社，1992

（孙强南　张广艳）

cunchu guanli chengxu

存储管理程序（memory manager） 操作系统中用于管理计算机主存储器及辅助存储器的程序。其主要功能有：地址转换、存储分配、存储保护和主存扩充。

地址转换 将程序地址空间中的逻辑地址转换成主存空间中相对应的物理地址。

存储分配 为进入主存的每道程序分配所需的主存空间。存储分配与主存区域的划分方式有关。一种是主存被划分成大小不等的连续区域，在这种方式下，存储管理采用的技术有：分区存储和分段存储管理，因而可采用分区或分段分配。分区方式使一个区域可以存放一个程序的连续的地址空间；分段方式使一个区域可以存放一个程序的逻辑分段的地址空间。另一种是主存被划分成大小相等的块，这时可采用页式存储管理，它将一个程序的地址空间划分成一连串的页面，然后，可被放置到主存不连续的存储块中。

存储保护 每道程序都在分给自己的主存空间中执行，保证互不干扰，以保护程序信息不被破坏和误用。常用的存储保护方法有：基址限长保护、上下界保护、存储键保护和存储环保护等。

主存扩充 虚拟存储器及其实现技术。基于程序局部性原理，将程序当前使用部分装入主存，其余部分存放在磁盘上，当被访问信息在主存中，执行就可顺利进行，如果被访问信息不在主存，为了继续执行下去，由操作系统自动将这部分信息从磁盘装入；如若此刻没有足够空闲物理空间，便把主存中暂时不用的信息移到磁盘上去。采用自动的“部分装入、部分对换”技术，用主存作为磁盘的高速缓存，让主存辅存独立编址但统一使用的技术，获得了一个容量与辅存相当，而速度接近主存的存储器。这种采用逻辑上对主存进行扩充的存储器，称**虚拟存储器**。常用的虚存管理方式有：页式虚存管理、段式虚存管理和段页式虚存管理。下面为页式虚存管理的基本概念和实现技术：

（1）页面　用户逻辑地址空间划分成若干等长的部分，每部分称为一个页面。页面从0开始依次编号，页面长一般为512字节至4096字节。

（2）页框　主存被划分成大小相同的存储块，每块称为一个页框。页框从0开始依次编号，页框与页面等长。

（3）逻辑地址表示　在分页系统中，每个逻辑地址用一对数（p，d）来表示。其中，p是页面号，d是在页号为p的页内的偏移地址，即页内地址。

（4）页表　描述页面虚实地址对应关系及页面和页框使用情况的登记表。页表表目中至少包含：页面号、页框号、外存地址和中断位等内容。

（5）快表　为了加快虚实地址转换过程，增加一个小容量联想存储器作为高速缓存。其中，存放程序执行过程中最常用的那部分页表称为快表。

（6）请求调页　当所需页面不在主存时，发缺页中断请求，暂停正在执行的程序，由系统分析中断源并从外存调入所需页面，修改页表后，让暂停的程序重新运行。

（7）页面替换　出现缺页且主存中无空闲页框时，淘汰主存中的一个页面并从辅存调入所需页面到相应页框。

（8）淘汰算法　又叫置换算法。它是当主存中空闲页框没有或过少，而又要调入新页面时，决定哪个页面从主存中移走的策略。常用的算法有：先进先出（FIFO）算法、最近最少使用（LRU）算法、时钟算法（clock）算法等。

分布式共享存储器（DSM）是一种存储器结构形式，它把物理上分开的存储器，逻辑上看成是一个大（共享）存储器，即这些物理上分开的存储器可编址为一个逻辑地址空间。分布式系统中或网络上的一组计算机能共享这个大存储器。利用DSM技术可以在无共享主存的分布式多机系统上运行基于共

享主存的并行程序。DSM 有基于页式、基于对象和共享变量的共享地址空间,其实现方法各不相同。由于可能有多拷贝,因此,需要通过一致性机制来保证多拷贝的一致性。

参考文献

孙钟秀,等. 操作系统教程.4 版. 北京:高等教育出版社,2008

（费翔林）

cunchuqi chacuo jiaoyan

存储器差错校验(memory error checking and correction) 对存储器中存储的信息进行正确性检测和纠错的方法和机制。差错校验建立在对信息字进行冗余编码的基础之上,即需要在原有的信息字中增加 1 位或多位,以增大编码后信息字之间的码距,使之具有检错或纠错的能力。通常,只有检错能力而无纠错能力的方案称作检验,既有检错能力又有纠错能力的方案称为校验。相应的,附加的冗余位称为检验位或校验位,原有的信息字和冗余位合在一起称为检验码或校验码。

在存储器中最常用的差错校验方案是奇偶检验和汉明校验。在奇偶检验中,检验码中只有 1 个检验位。若检验码中“1”的个数为奇数,称为奇检验码;若检验码中“1”的个数为偶数,则称为偶检验码。例如:

信息字	奇检验码	偶检验码
11110000	(1)11110000	(0)11110000
01110000	(0)01110000	(1)01110000

上例中,处于检验码最左侧括号中的那个二进位就是新增的检验码。对奇检验,检验位等于信息字中诸二进位的异或非;对偶检验,检验位等于信息字中诸二进位的异或。奇偶检验的实现分为两个阶段:①生成检验码。当信息写入存储器时,先生成奇检验码(或偶检验码),再写入存储器。②检验。当从存储器读出检验码时,检查它是否仍为奇检验码(或偶检验码),如果是,则存储正确,数据可用;否则报错。

奇偶检验只能检测出 1 位错的存在,但不能确定出错的具体是哪一位,因而不能纠错。汉明码设置了多个校验位,是一种可以检测出 2 位错并纠正 1 位错的校验码。它的基本思想是使每一信息位参与多个不同的奇偶检验组,在安排恰当时,这多个奇偶检验组通过重叠交错可以唯一地反映出校验码的出错情况。通常,为了纠正 N 位信息字中的 1 位错,至少应增加 K 个校验位,以指明 $K+N$ 位校验码中任一位出错的状态(共 $K+N$ 个状态)及没有出错的状态(共 1 个状态),即 K 应该满足 $2^K \geqslant N+K+1$。

现以 $N=10$ 为例加以说明。显然,此时 $K\geqslant 4$。令 10 位信息字为 $a_i(i=1,2,\cdots,10)$,4 位校验位为 $p_i(i=1,2,3,4)$,将这个 14 位的校验码按如下规则排列和分组:①校验码的位序从最低位 1 排到最高位 14;②从低位到高位,序号为 2 的幂次的位(即 1、2、4、8 位)放置校验位,其余放置信息位;③位序号为 $2^i(i=0,1,2,3)$ 的校验位参加由位序号的二进制表示中第 i 位(最低位是第 0 位)为 1 的信息位组成的奇偶检验组(如位序号为 $2^2=4$ 的校验位 p_3 参加的奇偶检验组包括 a_2, a_3, a_4, a_8, a_9, a_{10}, 因为它们的位序号 6, 7, 8, 12, 13, 14 的二进制表示中,第 2 位都是 1)。图 1 给出了具体的排列形式和分组情况,虚线框中的 1 代表参与某组,0 代表不参与某组。

下面以偶检验为例,给出各个校验位 p_i 的生成公式(采用奇检验的情况类似):

第 0 组　$p_1=a_9\oplus a_7\oplus a_5\oplus a_4\oplus a_2\oplus a_1$

第 1 组　$p_2=a_{10}\oplus a_7\oplus a_6\oplus a_4\oplus a_3\oplus a_1$

第 2 组　$p_3=a_{10}\oplus a_9\oplus a_8\oplus a_4\oplus a_3\oplus a_2$

第 3 组　$p_4=a_{10}\oplus a_9\oplus a_8\oplus a_7\oplus a_6\oplus a_5$

根据这样的分组,在检验的阶段,就可以由 4 个检验组的出错情况,唯一判定校验码中的错误位。例如,第 1 组和第 2 组出错,而第 0 组和第 3 组没错时,可以排除第 0 组和第 3 组的成员,即位序号二进

校验码	p_5	a_{10}	a_9	a_8	a_7	a_6	a_5	p_4	a_4	a_3	a_2	p_3	a_1	p_2	p_1
位序号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
第0组 S_0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1
第1组 S_1	0	1	0	0	1	1	0	0	1	1	0	0	1	1	0
第2组 S_2	0	1	1	1	0	0	0	0	1	1	1	1	0	0	0
第3组 S_3	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0
第4组 S_4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

图 1　$N=10$ 时的汉明编码矩阵

制表示中最高位为 1 的所有位置和最低位为 1 的所有位置,并锁定中间两位均为 1 的那个位置为错误位,即第$(0110)_2=6$位 a_3 出错了。一般地,有了 4 个检验组的检错结果,就可统一根据图 1 中虚线框内各列中二进位排成的数字,直接标定出错位置。令 $S_i(i=0,1,2,3)$为第 i 组的检验结果,即:

第 0 组　$S_0=p_1\oplus a_9\oplus a_7\oplus a_5\oplus a_4\oplus a_2\oplus a_1$

第 1 组　$S_1=p_2\oplus a_{10}\oplus a_7\oplus a_6\oplus a_4\oplus a_3\oplus a_1$

第 2 组　$S_2=p_3\oplus a_{10}\oplus a_9\oplus a_8\oplus a_4\oplus a_3\oplus a_2$

第 3 组　$S_3=p_4\oplus a_{10}\oplus a_9\oplus a_8\oplus a_7\oplus a_6\oplus a_5$

则由 $S_3S_2S_1S_0$ 组成的二进制数就指出了校验码中的错误位(组合 0000 表示没有错),据此就可以进行纠错(变 0 为 1 或变 1 为 0)。

为了检测两位错的存在,需要再增加一个校验位和一个奇偶检验组,即图 1 中的 P_5 和第 4 组。P_5 参与由所有 14 个数据位的奇偶检验,即:

$$p_5=a_{10}\oplus a_9\oplus a_8\oplus a_7\oplus a_6\oplus a_5\oplus a_4\oplus a_3\oplus a_2\oplus a_1\oplus p_4\oplus p_3\oplus p_2\oplus p_1$$

$$S_4=p_5\oplus a_{10}\oplus a_9\oplus a_8\oplus a_7\oplus a_6\oplus a_5\oplus a_4\oplus a_3\oplus a_2\oplus a_1\oplus p_4\oplus p_3\oplus p_2\oplus p_1$$

此时,如果第 4 组没错($S_4=0$)而第 0, 1, 2, 3 组中有错,说明检测到了两位错。

参考文献

王爱英. 计算机组成与结构. 4 版. 北京: 清华大学出版社,2007

(王玉祥　唐志敏)

cunchuqi leixing

存储器类型(memory type)　按照一定的标准将存储器分成不同的类型。存储器有多种分类方法。可按存储器在计算机中的作用及位置分类、也可按存储器的存取方式分类或者按存储介质分类等。

按存储器在计算机中的作用及位置分类,存储器可以分为**主存储器**(即内存储器)、**辅助存储器**(即外存储器)、缓冲存储器等。主存储器用来存放计算机运行时随时需要使用的程序和数据,它的工作速度较快,存储容量较小,主要采用半导体存储器,按随机存取方式工作。辅助存储器是一种不直接向中央处理器提供程序和数据的大容量存储器,它的工作速度慢,存储容量大,主要采用磁表面存储器和光存储器,按串行存取方式工作。缓冲存储器是位于两个不同工作速度的部件之间的、起缓冲作用的存储器。例如**高速缓冲存储器**、先进先出缓冲器(FIFO)等。

按照存储器的存取方式分类,存储器可以分为随机存取存储器(RAM)、**只读存储器**(ROM)、顺序存取存储器(SAM)、直接存取存储器(DAM)以及相联存储器等(CAM)。随机存取存储器(RAM, random access memory)是一种根据随机给定的地址对存储单元进行读写的存储器,每一存储单元的存取时间都是一样的,与存储单元在存储器中的位置无关。RAM 又可分为动态随机存取存储器(DRAM, dynamic RAM)和静态随机存取存储器(SRAM, static RAM)两种。DRAM 中存储的信息若长时间不被访问的话会自动发生变化,因此需要定时对其进行刷新来维持信息稳定;SRAM 在加电情况下所存储的信息能够稳定保持。只读存储器 ROM(read only memory)也采用随机存取方式。但是在正常工作时,ROM 只能读不能写;写入操作需要通过特殊的手段完成。按照写入方式不同 ROM 又可分为:可编程只读存储器(PROM, programmable ROM),可进行一次编程(写入)操作;可擦除可编程只读存储器(EPROM, erasable programmable ROM)可进行多次的擦除及编程操作;电可擦除可编程只读存储器(EEPROM, electrically erasable programmable ROM),工作原理与 EPROM 类似,但采用电擦除技术,因此速度更快。顺序存取存储器(SAM, sequential access memory)中信息的读写顺序完全按照其在存储介质上的物理顺序进行,存取时间与信息所在位置有关,典型设备为**磁带存储器**。直接存取存储器(DAM, direct access memory)的工作原理与 SAM 类似,其读写操作分两步进行:首先快速定位到信息所在位置的一个小区域,然后在此小区域内按照顺序对信息进行准确定位。DAM 的存取速度介于 RAM 和 SAM 之间,也被称为半顺序存取存储器,典型设备为**磁盘存储器**或光盘存储器。前面几种存储器都是按地址访问的,而相联存储器(associative memory)是按照存储单元中存放的全部或部分内容进行访问的,因此也被称为按内容访问存储器(CAM, content addressed memory)。

按照存储介质分类,存储器可以分为**半导体存储器**、**磁存储器**、**光存储器**等。半导体存储器是一种以半导体电路作为存储媒体的存储器,按其制造工艺又可分为 MOS 型存储器和双极型(TTL)存储器。MOS 型存储器具有集成度高、功耗低以及存取速度较慢等特点;双极型存储器具有存取速度快、集成度较低以及功耗较大等特点。磁存储器在金属或塑料基体的表面上涂抹一层磁性材料作为记录介质,工

作时磁层随载磁体高速运转，用磁头在磁层上进行读写操作。根据载磁体形状的不同，可分为**磁带存储器**和**磁盘存储器**。磁表面存储器具有容量大、位价格低以及存取速度慢等特点。磁盘常被用作辅助存储器，磁带则常被用作离线存储器。光存储器是一种利用激光进行信息读写的存储器，可分为只读式、一次写入式和可改写式三种，其最大的特点是存储容量大，是目前广泛使用的辅助存储器。

参考文献

1. 唐朔飞. 计算机组成原理. 2 版. 北京：高等教育出版社，2008

2. 白中英. 计算机组成原理. 4 版. 北京：科学出版社，2007

3. 王爱英. 计算机组成与结构. 4 版. 北京：清华大学出版社，2007　（郑衍衡　陈妍　王换招）

cunchuqi xingneng

存储器性能（memory performance）　表示存储器的工作速度（即存取时间）和存储容量的技术指标。存储器容量通常用能够保存的字节数来表示。为了表示更大的容量，常采用 k、M、G 和 T 等符号。例如，在以字节（B 或 Byte）为单位的计算机系统中 1 kB = 2^{10} B，1 MB = 2^{20} B，1 GB = 2^{30} B，1 TB = 2^{40} B。

主存储器的存取速度通常用存取时间和存取周期来表示。***存取时间***（memory access time，MAT）也称为访问时间，是指启动一次存储器读/写操作到该操作完成所需要的时间。***存取周期***（memory cycle time，MCT）是指连续启动两次存储器操作所需的最小时间间隔。由于存取周期还包括线路的恢复时间，因此存取周期略大于存取时间，其单位为 ns。

主存容量是指主存储器可以容纳的二进制位总数，高速缓存 Cache 是透明的，并不影响主存的容量。主存容量一般用字节为单位来表示。单个计算机的主存容量一般为数 GB 到数十 GB。

由于外部存储器（参见**辅助存储器**）的读写机构常常带有机械运动装置，它的存取时间与其保存数据的形式和机械装置的特性有关，用寻址时间和数据传输时间这两个时间参数来衡量。以硬盘（参见**磁盘存储器**）为例说明其存取时间的构成。硬盘上的信息是以同心圆的形式分布的，每一个圆称为一个磁道，信息以二进制位的形式沿磁道连续分布。硬盘的平均寻址时间包括平均寻道时间和平均等待时间两部分。平均寻道时间主要与磁头沿磁盘半径方向的运行速度有关；平均等待时间主要与磁盘旋转速度有关。硬盘的数据传输时间由其数据传输速率决定。数据传输速率 = 磁道容量 × 转速。在磁道容量相同的条件下，转速（单位为 r/min，RPM）越快，硬盘传送数据的速度也就越快。目前，普通 IDE 或 SATA 硬盘的常见转速为 5400 r/min 和 7200 r/min，服务器中使用的 SCSI 或 SAS 硬盘的常见转速为 10 000 r/min 和15 000 r/min。

硬盘的容量由磁盘的道密度和位密度决定。硬盘半径方向单位长度内的磁道数量称为道密度；沿磁道单位长度上的二进制位数称为位密度。磁道容量 = 位密度 × 磁道周长。硬盘容量分为非格式化容量和格式化容量。非格式化容量是磁记录表面可记录的磁化单元总数：非格式化容量 = 记录面数 × 每面磁道数 × 磁道容量。格式化容量是指按照某种特定的记录格式所能存储的信息总量，即用户可以真正使用的容量。格式化容量小于非格式化容量。目前常见硬盘的容量从数百 GB 到数 TB。

（郑衍衡　陈妍　王换招）

cunchuqi zucheng

存储器组成（memory organization）　有层次结构的存储系统的组成及各层次的存储器的功能。存储容量和存取时间是存储器的两个基本技术指标。**中央处理器**的高速运算要求存储器能在很短的时间内完成指令和数据的存取操作。由于高速存储器的价格昂贵，因此存储系统通常由**高速缓冲存储器**、**主存储器**和**辅助存储器** 3 个层次组成。高速缓冲存储器的存取时间最短，但容量最小。辅助存储器的存取时间最长，容量最大。在**操作系统**的管理之下，依赖于程序执行过程中的局域性特性，由这 3 个层次的存储器所组成的存储系统可以提供接近于高速缓冲存储器的速度和相当于辅助存储器的容量。

存储器中存放信息的基本单位是二进制位，用于存放二进制位的基本器件（或电路）被称为存储元或者存储基元，它是存储器中最小的存储单位，但是不能单独存取。将若干个存储元组成一个存储单元，它是具有特定存储地址的存储单位，存储器就是由很多这样的存储单元所构成。存储单元中所包含的二进制位数称为存储字长，通常存储字长等于机器字长。存储器可以按字编址或者按字节编址。若按字编址，则存储单元中的所有二进制位作为整体被写入或读出；若按字节编址，则可以对存储单元的全部或者部分字节进行读/写操作。

主存储器基本组成如图 1 所示。存储体通常是由存储元构成的存储阵列，是存储单元的集合。寻址系统由存储器地址缓冲寄存器(MAR)、地址译码器和驱动器组成，地址译码器接收到来自 MAR 的 n 位地址后，经过译码和驱动形成 2^n 个地址选择信号，每次选中一个存储单元。控制电路根据**中央处理器**(CPU)发出的读写命令，产生存储器内各部件需要的时序控制信号。读写系统由读写电路和存储器数据缓冲寄存器(MDR)组成，MDR 用来缓存 CPU 送来的数据，或者从存储单元读出的数据。读写电路读写控制信号，将所选中存储单元的信息从存储体读出暂存于 MDR 供 CPU 使用，或者将来自 CPU 并缓存在 MDR 的信息写入存储体被选中的存储单元中。

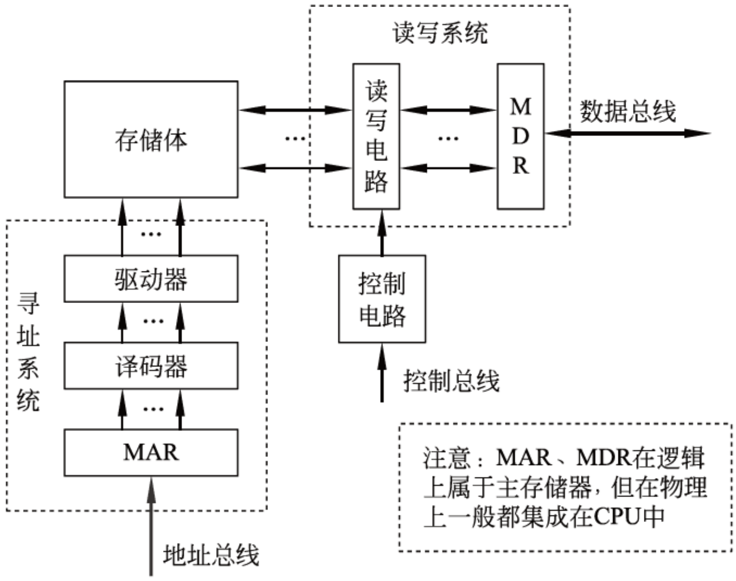


图 1　主存储器的基本结构

为了提高 CPU 访问主存储器的速度，目前存储器大多采用多体并行结构。主存储器由多个存储体构成，每个存储体及其读写电路和缓冲寄存器构成一个存储模块。这些存储模块可以实现重叠和交叉存取，因此可以同时从存储器取出多条指令或多个数据，从而提高计算机系统的运行速度。

存储器性能是影响计算机系统整体性能的主要因素之一。计算机系统对存储器的基本要求是高速度、大容量和低价格，但是这三个方面的要求之间存在矛盾。为了解决容量、速度和价格之间的矛盾，基于程序和数据的空间与时间局部性特点，通常把各种不同存储容量、不同存取速度和不同位价格的存储器，按一定的体系结构组织起来，形成一个多层次的存储系统，如图 2 所示。距 CPU 越近的存储体速度越快、容量越小且价格越高；距 CPU 越远的存储体容量越大、速度越慢且价格越便宜。较快存储体中数据是较慢存储体中数据的副本，数据一致性维护由硬件或操作系统自动完成，对系统程序和应用程序透明，存储系统总体效果可使程序员按大存储体的容量编程，CPU 执行时又可获得高速存储体的

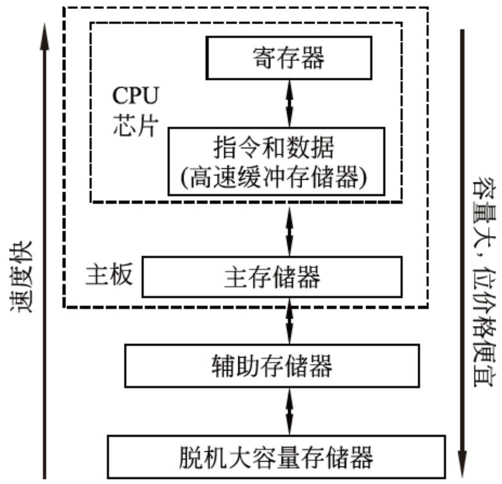


图 2　数据存储系统层次结构

读写速度。寄存器速度与运算器相当，容量最小。**高速缓冲存储器**(Cache)位于 CPU 和主存之间，其目的是提高 CPU 的访存速度；由 Cache 和主存构成的层次系统叫作 Cache 存储系统，其目标是提高速度，可以获得近似 Cache 的存取速度，一般通过硬件实现管理，对所有程序员透明。**辅助存储器**是容量

最大且位价格最低的存储器。由主存和辅存构成的层次系统叫作虚拟存储系统，其目标是扩充主存储器容量，一般通过操作系统实现管理，对应用程序员透明，使得应用程序员可以使用比实际主存地址空间大得多的虚拟地址空间进行编程。多层次存储系统能达到速度快、容量大、位价低的优化效果。

参考文献

1. 唐朔飞. 计算机组成原理. 2 版. 北京：高等教育出版社，2008

2. 白中英. 计算机组成原理. 4 版. 北京：科学出版社，2007

3. 袁春风. 计算机组成与结构. 北京：清华大学出版社，2011

（黄震春　陈妍　王换招）

cunchu quyuwang

存储区域网（storage area network，SAN）

一种连接外接存储设备和服务器的架构。它通过可伸缩的网络拓扑结构互连不同类型的存储设备与服务器，提供内部任意节点间的多路可选择的数据交换，并将存储管理集中在相对独立的区域内，实现最大限度的数据共享和优化管理以及系统的无缝扩充。

由于信息量快速增加，许多应用从计算密集型向输入输出密集型转变，原有以服务器为中心的存储方式无法满足网络环境中用户数据请求对输入输出性能的要求。而典型存储系统（如并行 SCSI 存储系统）存在着可扩展性差等问题。于是人们沿用了大型机专用输入输出系统中的存储网络概念，并利用高速串行的光纤通道（FC）接口技术，构造了 SAN。SAN 可以看作是服务器的专门负责存储的“后台”网络，处理面向块设备的输入输出操作，如图 1 所示。它也可看作是一种扩展了的共享存储总线，使存储设备不专属于某一特定的服务器。

SAN 由一个提供物理互连的通信体系和一个管理层构成。由于最早提出的 SAN 概念是基于光纤通道技术的存储设备网络，并且市场上的产品大都是光纤通道存储区域网（FC SAN），使得人们误以为 SAN 即 FC SAN。事实上，其通信体系可以利用其他的串行 SCSI（参见**外存储设备接口**）技术，如 SSA，ESCON，HIPPI 以及各种新兴网络技术，如**快速以太网**、InfiniBand 网络等。管理层负责对各个互连的存储设备进行组织，以确保数据传输的安全性和鲁棒性。

大多数存储区域网使用 SCSI 接口协议进行服务器和存储节点之间的通信。它们采用其他底层网络通信协议作为镜像层来实现网络连接，如光纤通道协议（FCP，Fibre Channel Protocol），一种最常见的通过光纤通道来映射 SCSI 的连接方式如下：

iSCSI，基于 TCP/IP 的 SCSI 映射；

HyperSCSI，基于以太网的 SCSI 映射；

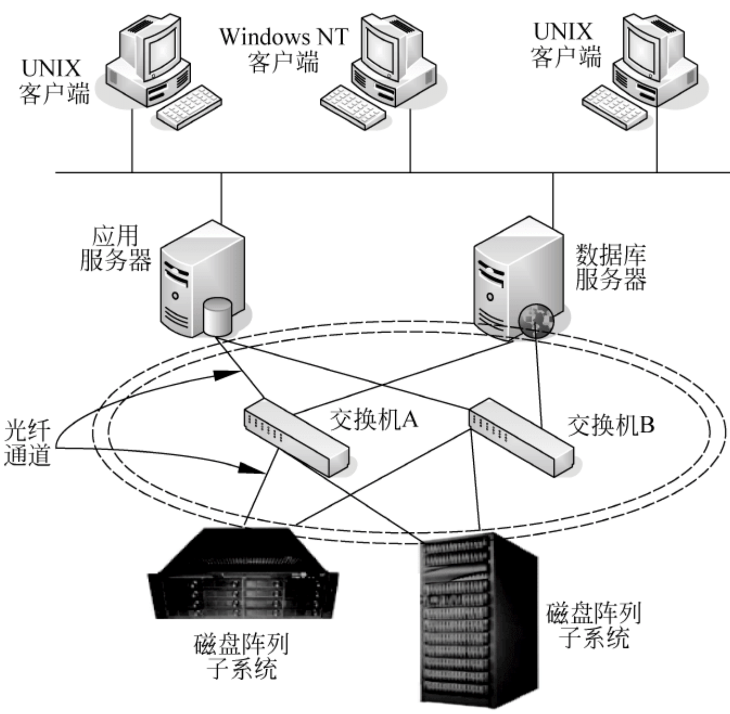


图 1　FC 通道连接构成 SAN 结构示意图

ATA over Ethernet,基于以太网的 ATA 映射;

Fibre Channel over Ethernet (FCoE),光纤连接的以太网;

iSCSI Extensions for RDMA (iSER),基于 InfiniBand (IB)的 iSCSI 连接。

采用 SAN 的优点在于:逻辑上数据是一体的,管理上集中控制,结构上易于扩充。SAN 有较强的容错功能,可充分利用网络带宽提高存取速度,因而可以保证数据传输的可靠性,并提供高性能、高可扩展、结构灵活的数据存储。基于 SAN 可以很好地实现磁盘镜像、备份与恢复、备份数据的存档与检索、不同存储设备间的数据迁移以及网络内不同服务器间的数据共享。SAN 通常与其他计算资源如 IBM S/390 大型机相毗邻,但也可以利用广域网技术,如异步传输模式(ATM)或同步光纤网(SON)延伸到远程进行远程备份和数据存档。SAN 适用于多种存储密集型应用领域,如非线性编辑、服务器集群、远程灾难恢复、因特网数据服务等。

SAN 的物理拓扑结构随所用网络技术而异,性能也随网络性能差异而不同。以 FC SAN 为例,按拓扑结构 SAN 分为三类。

(1) 点到点 SAN 具有网络远距离传输特性,能连接相隔 10 000 m 的远程存储设备,其远程连接能力不是传统总线所能具备的。连接配置中包括一发送端(光纤卡)和一目标端(可以是存储系统的光纤通道端口),无中继最远连接距离在采用铜线连接端口时为 30 m,采用短波光信号连接方式时为 500 m。

(2) 环形 SAN 是一种类似令牌环的共享带宽方式的拓扑结构。仲裁环路中至多可配置 126 个设备,设备间的通信通过仲裁以独占带宽方式进行。当环路中两设备间的通信完毕后才将控制权交给其他节点。每个节点都与其他节点共享一个单环带宽,双环路结构的带宽是单环带宽的两倍。环路连接一般通过光纤集线器来实现,环上的每个设备都会映射一个地址。当节点加入或从环路中移走时,环路会进入停止状态,并通过环初始化进程(LIP)重新分配相关节点的地址。环型 SAN 连接简单,但由于单存储设备故障将导致整个环路失效,系统安全性不高且不易管理。对于安全性和性能要求很高的应用,一般不适合采用光纤通道集线器连接方式。

(3) 交换式 SAN 结构中存储节点通过 FC 交换机与其他节点进行一对一的通信,每对节点间的连接带宽为光纤通道带宽。理论上,交换式光纤网络可通过级联扩充方式容纳 1600 万个节点。交换式 SAN 要求存储设备的光纤通道端口和光纤卡端口必须具有光纤登录能力。当端口登录到 FC 交换机时,经过信息交换,得到交换机分配给它的地址,同时登录信息将被注册到交换机中的单一名服务器表中。当节点加入和移出交换式 SAN 时,则不会产生环路初始化问题。如果两个节点间存在多于一个通道的连接,则当一个通道因故障不能进行通信时交换机将改变数据体的原始路径,以“路由”方式进行通信。

SAN 最初的产品主要是 FC SAN,FC 具有高传输速率并且该 FC 网络只用于存储,独享带宽,因而数据传输速率稳定。但是,FC 设备价格昂贵,不同厂家产品兼容性差,需架设单独网络设施,运行维护成本高,又因 FC 网络不能运用一般网络传输协议,影响了它的普及和使用。与此同时,价格便宜的以太网技术飞速发展,并在以太网技术基础上出现了基于 IP 协议的 IP SAN。IP SAN 利用 TCP/IP 网络协议的易访问、易管理、技术较成熟等特点,在保持传输速率高且稳定的基础上,大幅度降低了成本。

实现 IP SAN 有三种可选技术方案。

(1) 互联网小型计算机系统接口(iSCSI) 互联网工程任务部(IETF)和存储网络产业协会(SNIA)共同支持的开放协议,它将 SCSI 数据用 IP 协议封装,使 SCSI 命令与数据能在 IP 网络上传输。

(2) FCIP 采用将 FC 数据封装在 IP 包中进行传输。由于数据经 FC 和 IP 两次封装,传输效率不高,但通过 FCIP 方式可以有效地与远距离 FC SAN 进行互连。

(3) iFCP 通过在 SAN 的终端设备上映射 IP 地址来引入 TCP/IP 协议。此协议要求 FC SAN 的终端节点上安装专门网关硬件,由该硬件将所管辖 SAN 发出的 FC 数据包进行 FC IP 协议转换和转发。光纤信道协议(internet fibre channel protocol,iFCP)可以作为 SAN 内部协议或 SAN 间协议。

SAN 的带宽不断提高并且结构也越来越复杂,SAN 的低层互连技术如 FC 的速度从 2 Gb/s 向 8 Gb/s、16 Gb/s 发展,以太网从 1 Gb/s 向 10 Gb/s、40 Gb/s 甚至更高发展,IB 互联网络也已经发展至 40 Gb/s 以及 56 Gb/s。IP 协议的引入可以较好地解决互操作性,但同时使 SAN 的拓扑结构越来越复杂,用户可以通过主机、交换机、局域网/广域网(LAN/WAN)和虚拟专网(VPN)等设备访问存储数据。复杂的网络拓扑结构使存储网络面临着特殊的

安全风险。在 SAN 的发展过程中，不仅要处理更快、更大、更方便的问题，还要处理更安全等新问题。

参考文献

1. 张江陵，冯丹. 海量信息存储. 北京：科学出版社，2003

2. Farley M. Building storage networks. 2nd ed. New York：Osborne/McGraw-Hill，2001

（王芳）

cunchu xitong

存储系统（memory system） 由计算机中的内存储器（即**主存储器**）和外存储器（即**辅助存储器**）组成的子系统。

计算机对存储系统有 3 个基本的要求，即存取时间短、存储容量大和价格（存储一位信息的平均价格）低。这 3 个要求是互相制约的，存储器的存取时间越短，存储一位信息的价格就越高；存储器的容量越大，存取时间就越长。根据所能达到的技术水平，仅用一种工艺技术做成的存储器不可能同时满足这 3 个基本要求。为此存储系统采用由小容量的**高速缓冲存储器**、**主存储器**和大容量的低速外存储器组成的层次结构，具有这种结构的存储系统称为*层次结构存储器*。

存储系统的层次结构如图 1 所示。图中从上往下，存取时间和容量依次增加，存储一位信息的价格依次越少，即高速缓冲存储器的存取时间最短，容量最小，每位价格最贵；海量存储器的存取时间最长，容量最大，每位价格最便宜。在这种存储系统中，高速缓冲存储器的高速可以弥补主存储器在速度方面的不足，而外存储器的大容量可以弥补主存储器在容量方面的不足，所以，具有层次结构的存储系统可以实现高速度和大容量，而且价格合理。

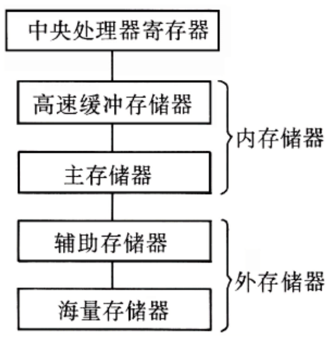


图 1　存储系统的层次结构

采用层次结构的依据是访问存储器的局部性。所谓访问存储器的局部性是指**中央处理器**访问存储器时，在一定时间内，无论是存取指令或存取数据，所访问的存储单元都趋向于聚集在一个较小的连续单元区域中。例如，程序一般都包含若干个循环段，当某一个循环段运行时，中央处理器就反复访问这个循环段中的为数不多的指令。又如，在进行向量、数组、表格等操作时，中央处理器也只是对聚集在一块的数据进行访问。访问存储器的局部性分为时间上的局部性和空间上的局部性。时间上的局部性指的是最近的将来要用到的指令和数据可能是现在正在使用的。空间上的局部性指的是最近的将来要用到的指令和数据在存储器中的位置可能和现在正在使用的相邻或相近。根据访存局部性，将中央处理器在近期使用过的指令和数据区域存放在高速缓冲存储器中，就可达到中央处理器高速存取指令和数据的目的。在程序执行过程中，高速缓冲存储器中的内容要随着中央处理器存取的指令和数据区域的变化而改变，即高速缓冲存储器中的内容要和主存储器中的某些内容进行交换；同样，主存储器的内容也可以成块地与辅助存储器进行交换。这些交换是计算机自动完成的，对程序员透明。这样，从中央处理器的角度来看，层次结构存储器具有接近于高速缓冲存储器的速度，同时又具有接近于海量存储器的容量。

在访问这种层次结构的分级存储系统中的某一级时，如果所存取的内容已经在这一级中，称为命中，否则，称为不命中。如果不命中，存储系统自动到下一级存储器中去寻找，并把该内容所在的区域交换到这一级存储器中。如果所需要的内容也不在下一级存储器中，则按同样方式到再下一级存储器中去寻找。提高命中率可减少各级存储器之间的数据交换，从而提高存储系统的效率。命中率和存储容量、所交换的数据块的映射策略、数据块的替换策略等有关（参见**高速缓冲存储器**）。

高速缓冲存储器中的指令或数据如果发生变化，主存储器中的副本应做相应的修改，使它们保持一致。这个问题在共享存储的并行计算机系统中变得更为复杂（参见**高速缓冲存储器一致性**）。

主存储器和辅助存储器之间的内容交换也是按块进行的。根据分块方式的不同，存储系统可分为页式存储系统（或称页式存储器）、段式存储系统（或称段式存储器）和段页式存储系统（或称段页式存储器）。在页式存储系统中，主存储器和辅助存储器都被划分为大小固定的页面，程序也被机械地划分为与页面大小相同的页，主存储器和辅助存储

器之间的内容交换按页进行。在段式存储系统中,程序被分解为多个可以明确定义的段,它们相互独立或基本独立,但又在逻辑上形成整体。段的长短不是固定的。由于程序运行时所需的地址空间大小随程序而异,段式存储系统更能体现访存局部性,但给存储管理带来困难。段页式存储系统将页式存储系统和段式存储系统结合起来,在段页式存储系统中,将主存储器分为页面,将程序分为段,每个段又分为若干个和主存储器页面同样大小的页,以页为基本交换单位。主存储器和辅助存储器之间的页或段的交换有多种策略,常用的是将近期最少使用的页或段交换到辅助存储器中去。

(郑衍衡 孙强南)

cunchu xunihua

存储虚拟化(storage virtualization) 一种通过映射或抽象的方式形成一个存储资源池,用屏蔽存储设备的物理复杂性,增加一个管理层面,激活一种资源并使之更易于透明控制的技术。全球网络存储工业协会(Storage Networking Industry Association,SNIA)对存储虚拟化给出两种定义:

(1) 通过将多个存储服务或存储设备集成的方法,来提供统一的存储设备或者存储服务,屏蔽系统的复杂性以及增加底层存储设备所不具有的新功能。

(2) 通过从应用、主机、一般网络资源中抽象、隐藏、隔离存储系统或者存储服务的内部功能,实现和应用、网络无关的存储管理。

网络存储虚拟化技术的发展大致可以分为3个阶段。

第一阶段(20世纪80年代中后期):这一阶段的存储虚拟化主要目的是扩大存储设备的容量和提高其性能,**磁盘阵列**通过将大量磁盘集中放置在一起,以容错的方式对数据块加以排列,避免因为单磁盘或双磁盘故障而导致数据无法使用。这类存储虚拟化通常附带在操作系统中,使得这类存储虚拟化具有极高的可配置性和灵活性,但是缺乏支持跨平台,作用范围比较小。此外镜像、分割和计算奇偶校验等任务占用了宝贵的中央处理器(CPU)资源和内存资源。

第二阶段(20世纪90年代中后期):**存储区域网**(SAN)的出现和应用对存储虚拟化提出了更高要求。这阶段通过共享存储资源以提高容量利用率,实现存储资源整合。在功能上,主要有简化管理,远距离复制数据,提高异地数据容错和对灾难事件的保护能力。SAN将存储虚拟化作用在更大网络范围的更多存储资源,但是在数据访问时,地址映射表所在的元数据服务器(参见**对象存储系统**)容易造成单点故障。此外,不同的存储虚拟化解决方案之间的数据迁移极其困难。

第三阶段(21世纪初期):SAN虽然改进了存储资源的利用率,但仍存在大量问题。比如不同供应商和设备之间互操作能力差,容易形成SAN孤岛。并且不同供应商的存储设备之间缺乏灵活性或复制能力。在这一阶段,存储供应商开始将存储虚拟化功能引入到自己的存储产品中。不仅提高了利用率,而且支持异构存储外部连接,提供不停机的数据迁移,实现业务连续性,此外还提供逻辑磁盘分区、分层存储以及精简预配置等功能。这一阶段,因为减少了一层管理,大大降低了网络复杂性,并且可以将虚拟化存储控制器的所有功能有效地扩展至外部存储,比如数据迁移和负载平衡,提高服务质量和安全性。

根据部署的位置可以将存储虚拟化分为基于服务器、基于存储网络和基于存储控制器三类。

基于服务器的存储虚拟化出现在服务器内部,最初是作为软件结合到操作系统中的,如今仍然非常流行。由于内置在系统软件中,这类存储虚拟化具有极高的可配置性和灵活性。因为包含在系统中,所以非常便宜,不需要配置其他硬件,并且可与操作系统识别的任何设备配合使用。但是只能以服务器为基础,作用范围比较小。由于镜像、分割和计算奇偶校验任务需要另行处理,占用了应用程序宝贵的CPU资源和内存资源。在数据迁移或复制时,整个环境的数据跟踪保护变得十分困难。

基于存储网络的存储虚拟化在网络层嵌入存储资源智能管理,抽象化服务器与存储阵列之间的实际存储资源。将磁盘同服务器分离,使得所有应用程序可以更有效地共享存储资源。这种虚拟化技术将异构供应商存储产品池化,形成一个可以无缝访问的存储池,从而可以在不相似的设备之间执行复制,并且提供单一管理接口。但是在数据访问时,由于需要一个映射表将虚拟地址重新映射成物理地址,可能成为单点故障影响使用效率。此外,不同的存储虚拟化解决方案之间的数据迁移极其困难,甚至根本不可能。

基于存储控制器的存储虚拟化允许其他异构供应商的存储阵列直接与自己的控制器连接,可以在

其基础之上增加外部存储资源,而且可按内部磁盘的相同方式进行管理。这意味着减少了一层管理,大大降低了网络复杂性,并且可以将虚拟化存储控制器的所有功能有效地扩展至外部存储。比如数据可在不停机的情况下在不同的存储池之间完成迁移或复制。此外,还可以通过分区,将端口、缓存和磁盘池等资源分配给特定的负载,以保持服务质量和安全性。

根据所处的位置,存储虚拟化可以分为带内和带外两类。带内方式,也称为对称方式,存储虚拟化设备嵌入在主机和存储设备的数据路径之中。主机在虚拟化存储设备执行 I/O,而不与真实的存储设备直接交互。所有 I/O 请求和它们的数据均通过存储虚拟化设备。带外方式,也称为非对称方式。存储虚拟化设备有时称为元数据服务器,它们只执行元数据映射的相关功能。主机通过访问元数据服务器获取数据的物理位置,然后直接和真实的存储设备交互。

存储虚拟化对企业有着切实意义,特别是在大型企业。基于存储控制器的存储虚拟化在兼容性方面具有独特的优势,各存储厂商正在不断研究和开发这类新产品。提供更大范围、更方便的异构存储设备接入方式一直是存储虚拟化中的核心问题,因此如何屏蔽异构设备的差异性仍然是未来存储虚拟化的研究热点。此外更大范围的异构设备接入对存储虚拟化技术的服务质量保证提出了更高的要求。结合分层存储技术,如何提高稳定、可靠的存储访问性能,更可靠的存储质量是存储虚拟化技术的又一个研究课题。

参考文献

张江陵,冯丹. 海量信息存储. 北京:科学出版社,2003

(李勇　施展)

cunchu yizhixing moxing

存储一致性模型(memory consistency model)　用以描述程序在执行过程中内存储器操作正确性的问题。内存储器操作包括读操作和写操作,每一操作又可以用两个时间点界定,即发出(invoke)和响应(response)。设系统内有多个处理器,假定单个处理器内指令是按顺序执行的,每个处理器可发出多个内存储器操作(读或写),那么总共有很多种可能的执行顺序。存储一致性模型描述的就是这些操作可能的执行顺序中哪些是正确的。

存储一致性模型是系统设计者与应用程序员之间的一种约定。如果应用软件遵从一定的规则访问虚拟内存系统,则应用软件可以获得正确的存储访问结果;反之,如果破坏了约定的规则,则存储访问的正确性得不到保证。

从某种意义上讲,存储一致性模型对共享存储系统中的多处理机的访存次序作了限制,从而对性能有一些影响。分布式共享系统的一个根本目标就是让一个通过**局域网**连接起来的工作站机群可以共享单一的虚拟地址空间,使之在工作站机群之上运行程序的效果等同于程序在单机上的运行。在最简单的变体中,每页存在于一个确定的机器当中,对本地页面的查询速度等同于对内存的访问速度,而访问远程机器的页面查询请求将引发段错误。这会使程序陷入到操作系统,由操作系统来处理缺页中断。操作系统随之发送一个消息至远程机器以找到所需要的页面,同时等待远程机器将该页面回送到本地。本地机器获得远程页面以后,引发段错误的指令会被重置,继续向下执行。为了达到这一目标,需要构造一个虚拟内存子系统,由它来捕获在分布式共享存储器(DSM)系统中的页面访问错误以及负责从网络上的其他节点处取回数据并完成必要的同步操作。

在前面所叙述的 DSM 系统中,某一个只读页面可能在多个节点上均存在副本,而对于可写页面,一般都由一个宿主来维护其一致性。远程访问的时候,由远程节点向宿主节点发出访问请求并从宿主节点处取回该可写页面。通常情况下,对一个可写的页面,DSM 系统中最多只能有一个副本。当应用程序的相关性比较大的时候,这种对每一可写页面仅仅维护一个副本的策略就会引发严重的性能上的瓶颈。维护多个副本的策略可以减缓性能上的瓶颈效应,但是又会带来新的问题,即如何在多个副本之间维护数据的一致性?为了解决这个问题,提出了存储一致性模型。

存储一致性模型的类别

按照一致性从强到弱的顺序,大体可以区别为下面几种类别。

1. 严格一致性模型　这是对一致性要求最严格的一种模型,它由下面的条件来描述:任何对内存位置 X 的读操作将返回最近位置对 X 进行写操作的值。

在 DSM 系统中,这是一种理想的模型,但是受到网络延迟的影响而不可能实现。在 DSM 系统中实现的多种一致性模型,都是对严格一致性模型在

不同程度上的放松。而在单机环境下,任何存储访问序列都满足严格一致性要求。

2. 顺序一致性模型　顺序一致性对存储器的限制比严格一致性要弱一些。顺序一致性的存储器要满足以下的条件:

(1) 每个进程的内部操作顺序是确定不变的;

(2) 假如所有的 CPU 上的进程都对某一个存储单元执行操作,那么,它们的操作顺序是确定的,即任一进程都可以感知到这些进程同样的操作顺序。

上述条件的含义是指当多个进程分别在不同的机器上并发执行的时候,只要所有的进程都保持同样的顺序访问存储器,那么,任何有效的交叉访问执行都是可以接受的。在顺序一致性模型中,时间不再是影响一致性的因素,它关心的是所有进程都必须能够感受到一致的内存访问序列。

顺序一致性模型不确保进程的一次读操作可以返回由另一进程所写入的最新值,在没有显式的同步操作的情况下,再一次运行同样的程序不能保证获得同样的结果。

3. 因果一致性模型　因果一致性模型是对顺序一致性的弱化,它要求在具有潜在因果关系的操作之间保持其一致的顺序。一般性的描述如下:有潜在性因果关系的写操作必须以同样的顺序被各个进程所感知,而并发的写操作在不同的机器上有不同的顺序。在基于因果顺序一致性模型的存储管理系统中合法的操作序列在顺序一致性或者严格一致性模型中可能会成为非法操作。

4. 管道一致性模型　管道一致性模型是在因果一致性模型上的进一步弱化,它满足下面的条件:由某一个进程完成的写操作可以被其他所有的进程按照顺序感知到,而从不同进程中来的写操作对不同的进程可以有不同的顺序。

管道一致性模型相对来说比较容易实现,所以在应用中具有一定的吸引力。实际上,它对于不同进程所感知到的写操作顺序并没有保证,除非是从同一个进程源来的写操作,则必须按照顺序到达别的所有的进程,类似于它们处在一条管道中一样。除此以外在管道一致性模型中,由不同进程产生的写操作完全是并行的。

5. 弱一致性模型　尽管管道一致性模型与较之严格的一致性模型相比,能够获得更好的性能。但因为它对同一个进程(源)所产生的结果仍然做了必须按序送达到所有别的进程的要求,所以在很多应用方面仍然受到了模型的限制。并非所有的进程都要求看到所有的写操作的结果,让他们按照顺序看到写操作的结果更是没有必要。这样处理,模型的开销会严重影响应用程序的性能。考虑到运算的中间结果在大多数情况下并没有“必须传送出去的必要”,我们对一致性要求作进一步的放松,修改为只有在需要传播写操作的结果的时候才将结果传送出去,除此之外,一切读写操作都完全是并行的。为了达到同步操作的目的,在弱一致性模型中引入了同步变量。

弱一致性模型必须满足的条件有下面几点:①对同步变量的访问满足一致性的要求;②对同步变量的访问,只有在以前的写操作在各处都完成之后才能完成;③对数据的操作(读或写),只有在以前的对同步变量的操作完成之后才能完成。

6. 释放一致性模型　对于同步变量的访问,弱一致性模型存在一个问题,就是无法区分进程是准备进入临界区还是已经完成对共享变量的操作而准备退出临界区,其结果就是进程在以下两种情况下都必须采取同步操作:①将局部写操作的结果传播出去;②从别的机器上收集共享数据的最新值。

如果将进入和退出临界区这两个动作区分,则可以实现一种更为高效的存储一致性模型——释放一致性模型。释放一致性模型提供了两类同步操作:Acquire 和 Release。某进程将要进入临界区时执行 Acquire 操作,退出时执行 Release 操作。也可以不用临界区而用栅栏(Barrier)同步来实现释放的一致性协议。栅栏是一种同步机制,它要求所有的进程全都到达程序的某一点后,各个进程才能继续往下执行。

借助于 Acquire 和 Release 操作,我们可以把某些特殊的共享变量保护起来,并维护它们的一致性。当然,应用程序必须知道它需要维护其一致性的数据,这也给应用程序的编制增加了一些协议开销,但是整体性能提高了。

通常,如果一个分布式共享存储系统满足释放一致性,则它必须遵守以下的规则:①某进程只有在成功完成 Acquire 操作之后,才能确保对一般共享变量(非共享同步变量)访问的正确性;②某进程只有在完成对共享数据的读写操作之后,Release 操作才能完成;③Acquire 和 Release 操作必须满足管道一致性要求。

7. 单项一致性模型　另外一种用于提高临界

区操作并行性的一致性模型是单项一致性模型。和释放一致性类似,它要求编程人员在临界区的开始和结束时使用 Acquire 和 Release 操作,但与释放一致性不同的是,它要求每个共享变量都与某同步变量相关联,同步变量可以是锁或者栅栏。以并行访问一个数组的不同元素为例,它要求给不同的数组元素加不同的锁,只有当对同步变量的 Acquire 操作完成之后,相关的共享数据才得到一致性保证。单项一致性模型也不同于懒惰释放一致性模型,后者并不将共享数据与锁或栅栏相关联,而是在对同步变量作 Acquire 操作之后,才能确定它需要哪些共享数据。

满足单项一致性的条件是:①在当前拥有者对数据的更新操作未完成之前,不能执行另一个进程对同步变量的 Acquire 操作;②如果某一个进程正以互斥模式访问某同步变量,则在该进程释放此同步变量之前,任何别的进程即使在非互斥模式下都将无法获得该同步变量;③如果某进程正以互斥模式访问一个同步变量,则在该进程完成操作之后,任何进程在以后对此同步变量的非互斥访问,都只有在成为同步变量的拥有者之后才能完成。

参考文献

吴俊敏. 存储一致性模型研究. 中国科学技术大学博士学位论文,2005

(张广艳)

D

dayinji ceshi

打印机测试(printer testing) 对打印机的质量与性能所进行的全面检测。打印机可按印字方式和印字处理技术分类。按照印字方式可分为串行式及并行式(行式),按照印字处理技术可分为击打式和非击打式。**击打式打印机**的主要代表是串行针式打印机,**非击打式印刷机**的主要代表有激光印刷机、喷墨印刷机等。

打印机测试主要包括打印速度测试、打印质量测试及打印功能测试。不同类型的打印机的测试方法也不尽相同。

打印速度测试 打印机的速度可用计时器进行测量。串行针式打印机的打印速度有平均打印速度和打印速度之分,测试过程中需明确打印机的工作状态。平均打印速度是测试打印机在连续打满行(包括换行)时,单位时间内能打印的字符数,单位为字符每秒或汉字每秒;打印速度测量则是测试打印机在脱机状态下其出针第一列到最后一列的总时间除以一行内的字符数,单位同上。激光印刷机由于采用页处理方式,其打印速度通常以页每分表示,测试时只需记录每分钟打印的页数即可。喷墨印刷机的打印速度同样也以页每分表示,测试方法与激光印刷机相同,但在测试时必须注明打印覆盖率,覆盖率是指一定幅面的打印媒体上,打印面积占幅面总面积的百分比,通常在5%覆盖率下进行测试。

打印质量测试 测试针式打印机的打印质量,首先要测量打印精度,打印精度测试可用专用仪器检测连续打印的横线或竖线,通过对成行度、成列度、走纸累积误差的测量可对打印机的打印质量有定量的测试。激光印刷机和喷墨印刷机的印刷精度是以dpi表示的,即每英寸可印刷的最多点数。分辨率指标在打印机的设计中就已定型,测试时只需编制一段程序,考查打印机是否在一英寸里能打印出相应的点数,也可在放大镜下观察打印机可打印的最小点的直径,考查其是否与标称值相符。另外,打印质量测试中还需考查打印机的打印文本清晰度,通常可通过打印一段不同字体和字号的文字,观察其清晰程度来考核此指标。

打印功能测试 针式打印机功能测试包括复制份数、噪声、纸处理功能、打印特性、接口、耗材、自检等的测试。针式打印机的复制份数测试主要考察打印头的击打力度。噪声测试可反映打印机的综合表现能力。此外,不同的打印机的纸处理功能、接口、耗材、打印特性、自检等功能会有不同的表现形式,在测试时只需对打印机进行自检或联机操作,若均能按各种命令打印出符合打印方式与质量的打印结果,则认为打印机合格。激光印刷机和喷墨印刷机的功能测试与针式打印机基本相同,只是无针式打印机的复制功能,因此也无须再测试复制份数。

如打印机需在特殊的环境条件下使用,还应根据有关国家标准或军用标准对打印机进行振动、冲击、高低温、湿度、电磁兼容、电源适应能力、可靠性等方面的测试。

随着打印机技术的发展,其测试方法也在不断地变化。其他类型的打印机如字模打印机、针式行式打印机、热敏式印刷机等,在具体的测试方法上可能会有不同,但总的来说可根据其类别参考上述几个指标进行测试。 (沈蓓)

daguimo bingxing chuli

大规模并行处理(massively parallel processing, MPP) 采用由大量(数百至数万甚至更多)处理单元构成的并行计算机系统,处理单元之间通过相互通信和协作,从而快速、高效地对大型问题进行求解的过程和技术。

发展简史

大规模并行处理的历史可以追溯到20世纪60年代的ILLIAC Ⅳ,这是一台由64个处理单元构成的单指令流多数据流(SIMD)计算机。早期的MPP系统都是SIMD型的,基于多指令流多数据流(MIMD)机制的MPP系统的实际可用性一直受到学术界的怀疑。1983年,美国加州理工学院研制了一台由64个Intel 8086/8087组成的基于**消息传递**的多计算机系统,称为Cosmic Cube。这一系统随后发展为Intel iPSC/1和iPSC/2。nCUBE公司在1986

年推出了 10 维超立方体系统 nCUBE/ten,该系统最多可配置 1024 个处理单元。1988 年,Sandia 实验室在 nCUBE 上计算 3 个大规模科学工程问题,得到了超过 1000 倍的加速比。这一工作首次从理论和实践两个方面展示了大规模并行处理的现实意义。

在 20 世纪 80 年代,在集中式**共享存储** MPP 系统方面也进行了大量的研究和尝试,如 IBM 公司的 RP3,纽约大学的 Ultracomputer,伊利诺依大学的 Cedar 等,但硬件的复杂性和性能上的不足限制了它们的进一步发展。80 年代后期以来,开始探讨在分布式存储器环境下实现共享存储的可能性。斯坦福大学的 DASH 就是一个典型的例子,而 Kendall Square Research 公司的 KSP-1 则是第一个商品化的分布式共享存储器 MPP 系统。

体系结构

鉴于集中式的共享存储器系统受存储带宽的限制和在实现时的复杂性,高度并行的计算机系统总是将物理存储器分散于各处理单元,以便为每个处理单元提供足够的存储器带宽,从而所有的 MIMD 型 MPP 系统在总体结构上都是相似的,即由处理单元结点和**互连网络**组成。每个处理单元结点可以是一个由**中央处理器**、**高速缓冲存储器**、局部存储器和互连网络接口组成的单处理机(当然结点还可以有输入输出接口)。不同系统的差别主要体现在网络接口中。例如:①早期的消息传递型多计算机系统采用存储转发(参见**路由机制**)的通信方式,互连网络采用超立方体结构;②后来的消息传递型系统在二维或三维网格上使用虫洞路由(参见**路由机制**)机制进行通信;③在**分布式共享存储**系统中,网络接口实现对远程数据的直接访问并维持数据的正确性;④在基于高性能工作站或 PC 的网络计算环境(亦称**集群计算**)中,网络接口实际上就是网卡。

为了避免消息传递型带来的低效率和不方便,并增强系统的通用性,有必要提供虚拟的逻辑上统一的地址空间。因此,出现了共享虚存和分布式共享存储器(参见**共享存储**)的概念,即在具有分布式存储器的系统中,通过统一的虚拟地址空间来实现存储器共享。

完全由软件实现的共享虚存适合在消息传递型系统和基于高性能工作站的局域网上实现。此时,每个处理单元的局部存储器被看作是整个虚拟地址空间的一部分,地址转换类似于典型的虚拟存储系统。只是缺页中断的处理过程有所不同,即新页面不但可能来自外存,而且更有可能来自其他处理单元的局部存储器。与此相反,在硬件实现的分布式共享存储器系统中,对非本地数据的访问是将远程数据由硬件自动取入本机高速缓冲存储器来实现。因为数据传输、共享和同步以及数据一致性操作的单位都缩小到了高速缓冲存储块,这种方式具有比共享虚存方式高得多的系统效率。

程序设计与软件环境

并行计算机的应用程序开发主要有两种形式,即串行程序的自动并行化和基于并行语言的并行程序设计。自动并行化的目标在于开发循环迭代间的并行性,虽已取得了一些进展,但效果仍不理想,且主要针对 FORTRAN 语言进行,适合于共享存储器的多处理机。基于并行语言的**并行程序设计**有多种方式,如共享变量、**消息传递**、数据并行、面向对象、函数式、数据流式等,但真正适合 MPP 计算机的主要是数据并行方式。

数据并行指的是将程序所要处理的数据域分割成许多小区域,从而可用区域的划分来代替计算的划分。这样,只要给每个处理机分配一个子区域,整个计算就能完全并行地进行。适合数据并行的应用都具有这样的特点,即对数据域中的每个元素都进行相同的计算,所以这类应用也适合在单指令流多数据流(SIMD)系统上实现。

数据并行模型具有单线程、并行操作于不同数据、松散同步、全局命名空间、隐式通信和隐式数据分配等特点。

尽管数据并行方式在一定程度上简化了在消息传递型多计算机上编程的难度,但仍然存在程序移植和并行化方面的困难。同时,数据并行只是数据结构的一种形式,且不能用于任务级的并行。消息传递方式编程时要为每个处理机单独编程,非常复杂,而且程序的伸缩性也不好。基于统一地址空间的共享存储器模型往往更适合于人们的思路,随着分布式共享存储器多处理机系统的进一步发展,基于共享变量的程序设计将逐步主导 MPP 系统的应用软件开发。虽然全新的非过程型语言在并行性的描述和开发方面有着 FORTRAN 语言和 C 语言无法比拟的优点,但实现效率低,并且大多和已有软件不兼容,它们难以占领市场。

MPP 计算机的系统软件实际上从事着与单处理机系统软件类似的管理工作,对大规模并行处理带来的新问题(如局部性、通信效率和负载平衡等问题)还缺乏有效的解决方案。纯粹依赖算法设计或程序设计来提高 MPP 系统的性能还不够。MPP

系统本身的并行化算法(即管理 MPP 计算机的算法)将更直接关系到并行处理系统的工作效率。

模型和方法

并行计算模型是 MPP 体系结构和 MPP 算法之间的界面,在这一界面的约定下,并行系统的设计者可以设计对并行性的支持机制以提高系统的性能;算法设计者可以开发高效率的计算方法以充分利用并行系统的计算能力。一个成功的并行模型不宜对硬件和软件结构的细节作过多的限制,从而保证它在相当范围内的通用性,便于算法和程序的移植;但它又要能很好地反映出不同结构的主要特征。

程序设计模型在很大程度上体现了计算模型。例如,顺序计算已有很成功的计算模型,即随机存取(RAM)计算机模型,它与传统计算机的冯·诺依曼结构是直接对应的。RAM 模型的并行版本是并行随机存取(PRAM)模型,它实际上对应着共享存储的编程模型。它的优点是特别适合于并行算法的表达、分析和比较,使用简单,易于设计算法,稍加修改便可运行在不同的并行机上,且有可能在 PRAM 模型中加入一些诸如同步和通信等需要考虑的问题。但由于 PRAM 模型对存储器的访问时间、存储体的分布和存储体的访问冲突未作任何限制,不能直接指导基于共享存储模型的并行程序设计。与消息传递型程序设计方式对应的是**通信顺序进程**(CSP)模型。在这个模型中,多个并发的进程通过互相传递消息进行通信与协作,尽管 CSP 模型具有良好的数学背景,有利于并行程序的正确性验证,但它本身不提供任何解决诸如通信开销、网络延迟等问题的手段。

LogP 模型对基于消息传递的并行系统中的开销给予了较多的重视,它用少量参数 L,o,g 和 P 来刻画并行机的主要瓶颈,这个模型的详尽程度足以反映并行计算设计时的主要问题,其简洁性也足以支持详细的算法分析,从而可作为 CSP 的一个重要补充,但 MPP 系统的可编程性仍未很好地解决。数据流模型便于程序设计,且具有彻底的并行性,但具体实现和效率上的困难经过近 30 年的努力仍未克服。

研究并行计算方法的主要目的是开发应用问题计算中潜在的并行性,从而使这种计算便于高效地在并行计算机系统上进行。并行算法研究的目标是把计算的时间复杂性尽量转化为空间复杂性,基本做法是加宽算法树的宽度,压低算法树的高度。合理的排序与分解(如区域分解、算子分解、数据分级等)有助于负载平衡和减少通信量,这是大规模并行计算提高效率的两个重要手段。

大量实践和分析表明,很多数学物理问题的潜在并行性体现在当规模增大时,并行加速比(参见**计算机性能评价**)呈线性增长。可伸缩性是研制并行算法的另一个重要指标。只有可伸缩的算法才能发挥可伸缩大规模并行计算机的高效率。适当的分块技术是实现可伸缩算法的重要手段,也是提高带有高速缓冲存储器的高性能计算机的性能所必需的技术。

存在的问题和发展方向

尽管 MPP 系统的峰值速度比向量多处理机高出 1 个数量级以上。但是 MPP 系统存在效率低、可编程性差、不同平台间并行程序的移植难度大等问题,而传统的向量超级计算机经过多年的发展,已积累了大量高效率的应用软件。其系统软件,尤其是向量化编译程序已基本成熟。因此,许多实际计算问题在这类系统上可获得很高的系统效率。从系统所能达到的持续性能出发,并综合考虑花费在算法改进和程序编制方面的人力和时间因素,MPP 的性能价格比有时比向量超级计算机的还差一些。

造成 MPP 系统的峰值速度与实际速度相差很大的因素很多,如局部存储器带宽不够,通信速度与计算速度不匹配,输入输出性能不能满足要求,系统软件效率不高等。MPP 系统大都基于高性能微处理器,而大多数高性能微处理器即使在片内高速缓冲存储器完全命中的情况下每个时钟周期也只能进行 1 次存储器访问,这与它们每个周期能执行多条指令或完成多次浮点运算的峰值速度不相称。通信速度与计算速度的不匹配,主要原因不在于传输速度,而在于通信本身的开销。

理想的 MPP 计算机系统在各方面的性能都应该是匹配和均衡的。这些性能包括处理单元的标量和向量计算速度、处理单元存储器容量和带宽、处理单元的通信速度和 I/O 带宽、互连网络的通信带宽和网络中半数结点同另一半进行通信的最大带宽,以及整个系统的计算能力和 I/O 能力。其中,由于技术方面的原因,I/O 带宽与运算速度的匹配最难实现。

MPP 计算机都有大量并行的计算单元,但大都缺乏大量并行的输入输出单元,因而输入输出成为系统最大的瓶颈。早期的消息传递型系统把 I/O 工作都交给系统主机去完成,造成了严重的瓶颈现象,也增加了通信网络的开销。为部分或所有处理

单元提供 I/O 能力和设备可以大大缓解这一瓶颈问题,但由此带来的资源管理工作(如并行文件系统等)仍是一个没有得到很好解决的问题。

传统单处理机迅猛发展的一个重要原因是其编译技术允许程序员用高级语言进行与机器结构无关的程序设计。相比之下,并行计算机(尤其是大规模并行计算机)的编译系统还远没有做到这一点。每个新的 MPP 机型的问世都面临着改写所有应用程序的艰巨任务。因此,为了进一步扩大 MPP 计算机的应用范围,必须实现与机器无关的 MPP 程序设计环境。

分布式共享存储模型可以方便地移植已有应用程序,便于开展自动并行化工作,因此具有一定的通用性。但是,它仍不能解决 MPP 系统实际速度远低于峰值速度的问题。相反,由于系统软件和优化技术在增强局部性、降低通信量等方面缺乏有效的手段,直接基于消息传递方式编程更利于发挥系统性能。消息传递和分布式共享存储模型都有一个不断完善的过程。

对于并行处理系统,下述几个部分是决定整个系统性能的关键:高速的单元处理机、存储系统、通信机制、同步机制和输入输出(I/O)系统。为了提高 MPP 系统的实际速度,一方面要改进这几个部分的设计,另一方面必须改善系统软件的效率,加强静态和动态优化的能力。例如,设法提高程序的访存局部性,有效地利用预取和后存技术,它们对提高 MPP 系统的性能有很大帮助。所有这些都需要体系结构、系统软件和算法设计三个方面的紧密配合。

参考文献

1. Gordon Bell. Scalable, parallel computers: alternatives, issues, and challenges. International Journal of Parallel Programming, 1994, 22 (1):3-46

2. 黄铠,徐志伟. 可扩展并行计算:技术、结构与编程. 陆鑫达,等译. 北京:机械工业出版社, 2000

（唐志敏）

daguimo shuju keshihua

大规模数据可视化(large data visualization)

对 TB 乃至 PB 量级数据进行分析并以直观的图形图像方式展现出来的技术,其主要策略是采用并行算法设计,合理利用有限的计算资源,高效地分析和可视化大规模数据集。

在面向大规模数据的并行可视化工作中,主要涉及四种基本技术:①数据流线化(data streaming)　将大数据分为相互独立的子块后依次处理。在数据规模远大于计算资源时主要的一类可视化手段。它能够处理任意大规模的数据,同时也可能提供更有效的缓存使用效率,并减少内存交换,但通常这类方法需要较长的处理时间,不能提供对数据的交互挖掘。离核渲染即是数据流线化的一种形式。②任务并行化(task parallelism)　把多个独立的任务模块平行处理。这类方法要求将一个算法分解为多个独立的子任务,并需要相应的多重计算资源。其并行程度主要受限于算法的可分解粒度以及计算资源中节点的数目。③管道并行化(pipeline parallelism)　同时处理各自面向不同数据子块的多个独立的任务模块。对任务并行化和管道并行化方法,如何达到负载的平衡是一个难点。④数据并行化(data parallelism)　将数据分块后进行平行处理,通常称为单程序多数据流(SPMD)模式。这类方法能达到高度的平行化,并且在计算节点增加的时候可以达到较好的可扩展性。对于非常大规模的并行可视化,节点之间的通信往往是制约因素。提高数据的本地性也可以大大提高效率。以上这些技术往往在实践中相互结合,从而构建一个更高效的解决方法。

可视化中图形的绘制是一个计算密集型的处理工作。在处理大规模数据时,使用可视化算法,以互动的速度来绘制图形已经超出了单一的 CPU 和 GPU 图形加速器的计算能力。数据并行绘制方法被普遍用于提高可视化系统的交互速度。最常用的并行绘制算法的分类是基于绘制流水线中图元排序的位置。依据排序的先后,可以大致分为:首排序(sort-first)、中排序(sort-middle)和末排序(sort-last)。

首排序(sort-first)算法在绘制流水线的起点分配基本图元,通过分割输出图像区域,给每一个处理节点分配相应的区域。一旦这些基本图形分配完成后,每一个处理器会完成整个图形管线的处理而生成最终的子图像。先排序算法可以充分发挥每个节点图形硬件加速器的性能,处理器间的通信要求低,从而能获得较低的开销和更高的性能。先排序算法的主要缺点是可能会造成工作量分配的不平衡。

中排序(sort-middle)算法中,数据的分配发生在绘制流水线的几何处理和扫描转换阶段。绘制流水线的分裂是中排序方法的最大不足,在早期,它很难充分利用图形硬件加速器进行绘制。但随着可编程图形硬件加速器的发展,这部分问题可以得到部

分解决。此外该方法也会有工作负载量不均衡。

末排序(sort-last)的方法把排序推迟到绘制流水线的最后阶段。基本图形的初始化分配是采用随机的方式,每一个处理器绘制其相应的最终图像。所有的这些子图最终复合成一个完整的图像。处理器间通过高速网络来满足交互绘制的需求。末排序的可以完全利用整个图形处理器的渲染性能,并能较好均衡工作负载。其主要的缺点是在图像合成阶段,需要发送大量的数据。其中,二分交换合成法(binary-swap compositing)较好地利用了图像合成中限制的计算和通信能力,是一种较为经典的方法。

近年来受到关注的一种针对模拟计算产生的大规模数据的可视化模式称为原位可视化(in situ visualization),通过模拟计算和可视化紧密结合,降低数据传输和存储的成本。如图1所示,在通常的可视化模式中,将PB量级模拟产生的全部数据传递到存储设备,再经处理后用于可视化。数据传输是整个系统的瓶颈,I/O将占据绝大部分的计算时间。而在原位可视化模式中,数据直接在计算后原位被缩减与前处理,再用于随后的可视化与分析。经过缩减后的数据,通常比原始数据小多个数量级,能够极大地降低数据传递和存储的开支。

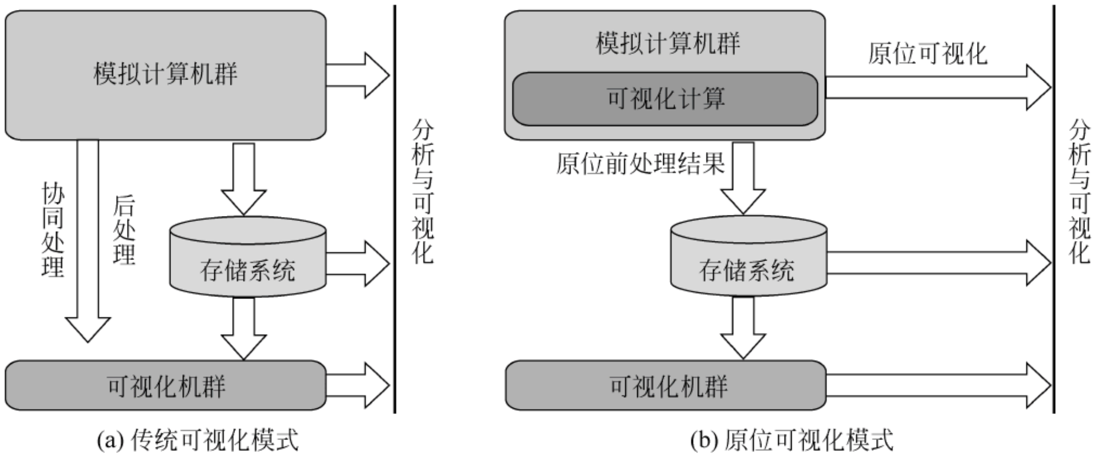


图1 两种可视化模式的比较

图形硬件对于大规模数据可视化具有重要意义。最新的超级计算机大量地应用GPU作为计算单元。如何更好发掘最新的图形硬件的潜力,提供更加灵活的大数据可视化和绘制的解决方法是具有重大意义的课题。除了科学计算数据外,也要关注信息可视化中大规模数据的涌现,研究此类工作的大规模分析处理方法将很快成为迫切的需求。

参考文献

1. Ma K L, Painter J S, Hansen C D, et al. Parallel volume rendering using binary-swap compo-siting. IEEE Computer Graphics and Applications, 1994, 14(4): 59-68

2. Ma K L. In Situ visualization at extreme scale: challenges and opportunities. IEEE Computer Graphics and Applications, 2009, 29(6): 14-19

3. Mccormick P, Ahrens J. Large-scale data visualization and rendering: A problem-driven approach. In: Hansen C D, Johnson C R (eds), The Visualization Handbook, Elsevier, 2005

4. Molnar S, Cox M, Ellsworth D, et al. A sorting classification of parallel rendering. IEEE Computer Graphics and Applications, 1994, 14(4): 23-32

(袁晓如)

daxing jisuanji

大型计算机(large computer, mainframe)

使用所在时代的先进技术、为频繁使用输入输出设备的高速数据处理而设计的一类**通用计算机**。它代表该时期计算机技术的综合先进水平。

大型计算机的处理系统可以是单处理机、多处理机或多个子系统的复合体。处理机一般采用两级以上**高速缓冲存储器**、流水线技术和多执行部件以提高性能。存储系统一般由高速缓冲存储器、**主存储器**、磁盘存储系统和海量存储器组成,构成多层级的存储系统。输入输出系统由通道和外围设备组成,通道专门负责输入和输出处理(参见**输入与输出通道**),一般有字节多路通道、选择通道、成组多路通道等。

大型计算机不断革新,在大型企业、政府机构和金融部门的关键业务数据处理中占据重要地位。

发展简史

20 世纪 60 年代中期以前的计算机都称为大型主机,各自为不同的应用(科学计算或数据处理)设计,具有不同的体系结构。近 50 年来大型计算机的发展大致可以划分为 4 个阶段:

第一阶段为 20 世纪 60 年代中期,以美国 IBM 公司推出的 System/360 计算机为代表,创立计算机体系结构概念的发展阶段。它是大型计算机向系列化、通用性、兼容性发展的典型,具有以下主要特征:其可扩充、可兼容的体系结构规定了硬件和软件之间界面的规范和实现的预期行为,不仅使该系列中各个机型具有兼容性(参见**系统兼容性**),而且使以后相继发展的大型计算机系列具有向后兼容性;具有标准的输入输出接口,使输入输出设备与中央处理机保持相对独立,并规范了 System/360 所有机型中均能使用的输入输出设备;System/360 采用混合集成电路为逻辑元件,计算机系统可用于数据处理和科学计算。在System/360 推出之后,世界上若干大型计算机厂商相继开发与 System/360 兼容的计算机和外围设备。

第二阶段为以 1970 年 IBM 公司宣布的 System/370 为代表的体系结构成熟阶段。System/370 体系结构扩充了指令集、引入虚拟存储器和成组多路通道技术。经过 System/370-XA 扩充体系结构后,发展成 ESA/370 体系结构,地址空间由 24 b 扩大到 31 b,引入数据专用的虚拟空间和使输入输出高速化的超空间。

第三阶段为 1990 年 IBM 公司宣布的 IBM System/390 为代表的体系结构革新阶段。ESA/390 体系结构的特点是在 ESA/370 基础上增加多子系统复合体 Sysplex,ESCON 光纤通道体系和共同密码体系 CCA 等,以适应网络计算环境和电子商务应用快速增长的需要。

第四阶段为 2000 年 IBM 公司发布 Z/Architecture 为代表的体系结构重大发展阶段。地址空间扩大到 64 b,若干新概念和新技术使再次革新的大型计算机具有数据密集型可变规模计算能力。2010 年宣布的 Z Enterprise System 196 成为速度最快、可扩展性最强、能源效率最高的新一代大型计算机。它与刀片服务器结合,利用统一资源管理软件,进一步扩大混合计算模式和增强 I/O 子系统能力。将大型计算机的高可靠性和安全性带入到大型数据中心的其他子系统中,成为大型数据中心的统一管理中枢。

应　用

大型计算机可扩充、可兼容的体系结构,大规模的 I/O 吞吐能力及其具备的高可靠性、高可用性和高服务性,使其在国防、金融、保险、交通、电信、商业等行业和领域的关键业务中的应用一直受到重视。全球 50 大银行以及 500 强企业中,大多依靠大型计算机执行复杂的事务处理。

大型计算机在军事领域,可用于全球性或区域性的战略防御系统、大型预警系统、航天测控系统等。在民用领域,可用于大型企业信息处理和管理系统、电子商务交易系统、银行业务事务处理系统、大区域中长期天气预报信息系统、大面积物探信息和资料处理系统、大型工程设计和模拟系统等。

目前,基于大型计算机的独立应用软件达 5000 多种,其中将近 2500 种基于 Linux。全世界有 1400 多家独立软件提供商在开发新一代大型计算机的应用,预示着大型计算机及其应用仍然具有较好的发展前景。

发展趋势

在计算机领域,随着**微处理器**与**互联网**的应用和普及,计算模式从集中计算模式向分布计算模式、并逐渐向分布-集中计算模式演变。大型计算机的角色从大型主机向高端服务器、进而向企业云计算平台发展。在**云计算**的推动下,新一代大型计算机将具备可变规模计算能力、共享资源能力及统一管理异质计算基础设施的能力。

随着大数据时代的到来,集成大交易数据和大交互数据、进行大数据处理和分析、了解和预测客户的行为和需求,并基于这种需求提供服务和产品,已成为一种新的趋势。这种大趋势将促进新一代大型计算机具备大数据处理、分析和管理能力,成为支持电子商务向智慧商务转变所必须的重要工具和手段。

随着计算机类型的多样化,不同场合应用不同的计算机类型,建立以大型计算机为中心的开放的混合计算环境将成为另一重要趋势。所谓开放的系统指在系统组成部分,特别是应用编程界面、用户交互界面和网络连接界面采取统一规范或标准。目前市场流行的开放技术,诸如 Java、Linux、Unix 等在大型计算机上都已获得支持。

大型计算机将继续降低系统成本,不仅是设备成本,还包括风险代价、系统维护、能源消耗、软件授

权、空间占用等众多因素的费用;还将降低每秒百万条指令(Mips)的耗电量,减少每次核心交易的能耗。随着多核微处理器技术和多芯片模块组装技术的改进和提高,大型计算机占用的空间亦将大为减少。

参考文献

Hennessy J L, Patterson D A. 计算机系统结构—量化研究方法. 3 版. 郑纬民,汤志忠,汪东升,译. 北京: 电子工业出版社,2004 (李经纬)

daxing jisuanji dianyuan xitong

大型计算机电源系统(power supply system for large-scale computer) 根据**大型计算机**对电源的容量、质量、控制、检测等要求,把众多的直流电源、交流电源设备用供配电装置、监控装置连接而成的供电整体。电源系统的可靠运行,是计算机稳定工作的必要条件。

电源系统的设备

静止变换式不间断电源 大型计算机担负着重要的控制、运算或管理任务,即使短暂的停机,也会造成重大的损失。所用电源能够利用蓄电池的储能,解决停电问题,并提高供电质量。

低电压大电流直流电源 计算机逻辑电路需要低电压直流电源供电。不同的器件需要不同的电压。每种电压需要总电流为数百到数千安。所以计算机硬件系统必须分割为规模适当的模块,使每个模块需要的电流适度,便于供电。大型计算机采用两类直流电源:多相整流电源(PRPS)和大功率开关电源(SMPS)。这两类电源具有输出功率大、效率高、体积小、可靠性高的优点。另外,为保证供电的安全性,大功率电源内部都设有过压、过流、过热保护。所以,一旦有异常情况出现,电源就会自动关断,对负载和电源本身进行保护。

交流配电柜 交流电经配电柜的分路开关接到各个直流电源的输入。配电柜中设置引出保护线的端子,还有检测和显示交流电压、电流、频率、功耗等参数的功能,而且当某一参数超出规定范围,就会发出声光报警,确保机房供电的安全性。

电源系统的构成

开关电源系统(图 1)和多相整流电源系统(图 2),是大型计算机采用的两种电源系统。在开关电源系统中,50 Hz 的**不间断电源**(UPS)和配电柜对开关电源供电,同时又对外围设备和前端机供电。在多相整流电源系统中,为缩小整流电源的变压器和滤波器的体积,交流供电配备了 400 Hz 的不间断电源和配电柜,专对多相整流电源供电。因为 400 Hz 的不间断电源不能用市电作为后备电源,所以为保证供电的连续性,400 Hz 的不间断电源必须有备用设备。比较两种系统,显然后者比前者设备多、投资大、交流供电复杂。另外,多相整流电源输出不稳压,只适用于功耗基本恒定的 ECL 器件。但它具有线路简单,调试维护方便,可靠性特别高等优点。

电源系统的可靠性

大型计算机对电源可靠性的要求很高,通常采取以下技术措施: ①交流电源采用两路市电供电,一路市电作为正常供电电源,另一路作为后备电源。这是提高交流供电可靠性的最有效,同时也是最经济的方法。②选用技术成熟的交流电源设备和**直流电源**。电源设备在实际使用中不能满负荷运行,要降低额定电流 30% 左右;并需改善环境条件,加强通风散热。③在特别重要的部位采用并联冗余结构。图 3 是 $N+1$ 冗余结构图,其中各电源性能完全

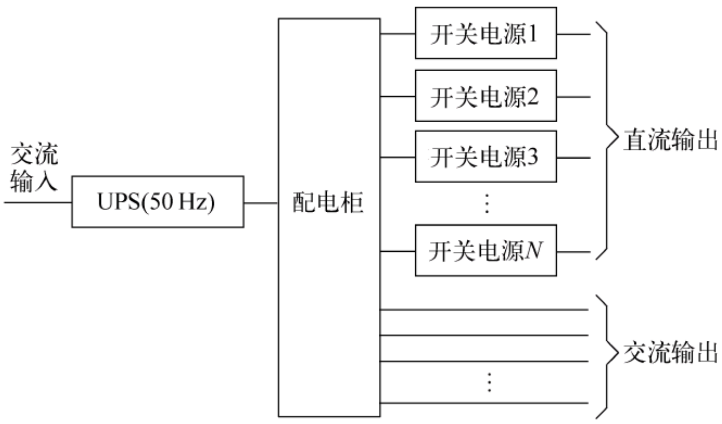


图 1 开关电源系统

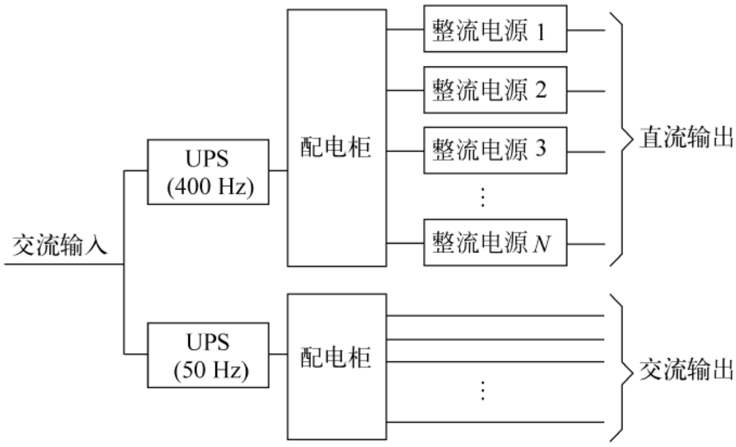


图 2　多相整流电源系统

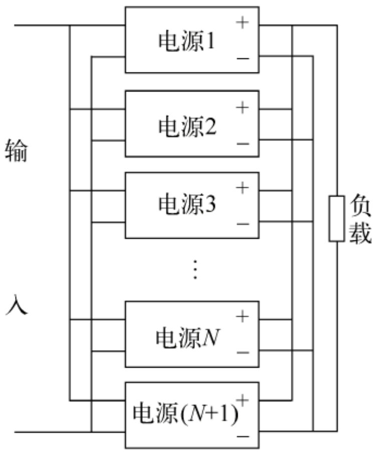


图 3　$N+1$ 冗余结构图

一致,输出端允许直接并联,并能自动均流。负载电流可由 N 台电源提供,而实际电源为 $N+1$ 台,冗余 1 台。工作过程中,万一有 1 台电源出故障,系统仍能正常工作。在及时更换故障电源之后,电源系统仍处于冗余状态。

电源系统的控制

大型计算机有很多电源,分布在各个机柜内,多设有电源监控装置。监控装置通过遥控来开关各机柜的直流电源。为了减小开关过程中的浪涌电流,并观察加电过程是否正常,电源可根据程序安排的次序自动逐台开关,直到整个系统开机或关机过程结束。监控装置对各台电源的输出电压进行定期的采样和检测。发现电压异常,即将有关硬件脱机,使系统的其余部分照常工作;同时进行声光报警,指示发生故障的部位,以便维护人员进行处理。监控装置也对不间断电源的工作状态进行监测。如果发现交流市电故障,不间断电源已经进入蓄电池维持供电状态,就及时发出报警信号,以便进行应急处理。

参考文献

王其英. 计算机电源系统的设计. 北京:科学出版社,1987　　(诸云龙)

daxing jisuanji huanjing kongzhi xitong

大型计算机环境控制系统(environment control system for large-scale computer)

用于满足**大型计算机**系统中的电子设备所处环境的空气温度、湿度和洁净度等环境指标要求,以维持其长期地、不间断地、稳定地运行而设置的各种设备、设施。

大型计算机系统环境控制系统技术的发展　主要跟随计算机设备的发展及其对环境要求的变化以及使用者对更高的节能性的追求而相应发展。例如早期大型计算机系统中,计算机设备多被设计成垂直通风的散热方式,所以早期的空调设施使用下送风式空调机,并配合高架地板送风的气流组织方式。近年来,随着大型计算机系统中计算机设备的机架化、刀片化,水平送风的空调机设备配合水平送风的气流组织形式的应用也越来越普遍。

大型计算机的环境控制系统　通常包括用来控制空气温度的制冷系统(或冷却系统)和控制空气湿度和洁净度的空气处理系统。制冷系统包含用来制冷的空调机设备和用来将冷气从空调机输送到计算机设备并返回的气流组织系统。空气处理系统通常包括空气处理机或单一功能的加湿机、除湿机、除尘机等设备。有些空调机设备同时具有制冷、除湿、加湿和除尘功能。常见的空调机有直冷型和冷冻水型两种类型,直冷型空调机直接利用一次冷媒(目前

多用卤代烃类化合物作为制冷剂,亦称氟利昂)利用蒸汽压缩循环直接对室内空气进行制冷,冷冻水型空调机则利用一次冷媒对二次冷媒(目前多用纯水或乙二醇的水溶液作为载冷剂)进行制冷,然后利用二次冷媒对室内空气进行制冷。常见的加湿机有电极式、远红外式和湿膜式,其中电极式和远红外式均通过加热方式使水蒸发,湿膜式则是通过气流使水在常温下蒸发。常见的除湿机有制冷式和化学式,前者利用低温使水蒸气结露而除湿,后者利用化学制剂吸收水分而除湿。常见的除尘机均采用空气循环、过滤的方式,降低空气的含尘量。常见的气流组织方式有高架地板送风循环方式和水平送风循环方式,有些情况下,为了约束气流的流向、减少气流在输送、返回过程中的流失,还采用送风管道进行送风或回风,或利用经封闭后的机柜排间物理通道或机柜排内专用风道进行送风或回风。

大型计算机的环境控制系统未来主要向以下几个方向发展:①适应更高的设备功率密度;②更加节能;③与计算机设备整合;④模块化。由于计算机设备的小型化以及**中央处理器**(CPU)性能的大幅提高,单机柜的功率密度也在不断上升,30 kW 的单机柜功率密度将很常见,从而对环境控制系统提出了相应的要求:水平送风、就近送风、温湿解耦、封闭通道、封闭风道等技术将被广泛应用;能够充分利用室外环境条件的空气自然冷却技术、河流自然冷却技术也将有所应用。为了进一步降低能耗、提高电能利用率或提高计算性能,环境控制系统将与计算机设备的内部散热系统整合,形成一体化的具有环境控制功能的计算机系统。出于对高可用性、快速部署、易扩展的需求,环境控制系统将被设计成为由多个可不中断修复或更换的、可简便安装的模块组成的系统。 (沈卫东)

daima shengcheng

代码生成(code generation) 把经语法语义分析后的中间结果转换成等价的目标程序或目标程序模块的过程和描述。中间结果是用三元式、四元式或逆波兰式等中间语言表示的程序。目标程序是用目标语言书写的。目标语言可以是**机器语言**、**汇编语言**乃至**高级语言**。随着计算机的编译环境和执行环境的不断完善,不仅能产生可直接执行的目标程序,也能生成可再定位的或可连接的目标程序模块,并最终通过连接装入程序构成可直接执行的目标程序。

代码生成不但与编译有关,而且与运行环境有关。所生成的目标程序应当符合连接装入所要求的规范。代码生成中存储分配和寄存器分配是直接影响目标程序功效的重要且复杂的问题,应给予足够重视。

参考文献

Aho A V, Sethi R, Ullman J D. Compilers principles, techniques and tools. Addison-Wesley, 1986

(钱树人)

daima youhua

代码优化(code optimization) 在语言处理过程中为提高程序质量而采用的技术。需求级、功能级和设计级语言处理系统的优化技术尚处于实验探索阶段,实现级语言处理系统尤其编译程序的优化技术发展较早、较为成熟。

编译阶段上可进行的优化分为中间代码级的优化和机器代码级的优化,又分别称为与机器无关的优化和与机器有关的优化。

中间代码级的优化有表达式优化、循环优化等。表达式优化通常有合并常量运算、提取公共子表达式、布尔表达式优化等。循环优化通常有外提循环不变式、强度削减等技术。

机器代码级的优化有寄存器优化、机器指令的有效利用等。

从涉及的范围上优化技术又可分为***局部优化***和***全局优化***。若优化涉及的范围是一个基本块,则称为局部优化,否则称为全局优化。

在优化工作中循环优化,尤其是与内循环有关的优化特别重要,它对程序的质量有显著影响。

优化技术的基础是数据流分析和控制流分析等。

目标程序的质量不仅与编译的优化技术有关,而且与源程序所采用的算法有关。求解一个问题的算法可能有许多,它们的复杂度常常会有数量级上的差异。采用高质量算法的源程序是提高目标程序质量的关键所在。

参考文献

Aho A V, Sethi R, Ullman J D. Compilers principles, techniques and tools. Addison - Wesley, 1986

(钱树人)

daishu guiyue

代数规约(algebraic specification) 主要用于

描述抽象数据类型的基于代数理论的形式规约方法。代数规约的数学基础是多类一阶等式逻辑。一个代数规约由基调和公理集两部分组成。基调给出所涉及的类别，以及所定义的函数符号的类型。这些函数所应具有的性质则由一组公理来刻画。公理的形式通常是等式或条件等式。

例　自然数的代数规约

NAT =

sorts nat

opns 0： →nat

S：nat →nat

eqns x + 0 = x

x + S(y) = S(x + y)

x * 0 = 0

x * S(y) = x + x * y

对一代数规约的每个类别给定一个载体集，将函数符号解释为与其类型相应的集合上的函数，就得到一个代数结构。如果规约的每条公理在此结构中都成立，就说该结构是规约的一个模型。一般说来一代数规约可有无穷多个模型。比如单元素结构（载体集只有一个元素 e，所有运算作用在 e 的结果均为 e），或者通常的自然数结构（载体集为{0，1，2，…}，常量符号 0 解释为数 0，函数符号 S 解释为后继运算. eps，+ 和 * 解释为通常的加法与乘法运算），都是上述例子中 NAT 的模型。NAT 还有无穷多个“非标准模型”。但是只要公理的形式是等式或条件等式（或更一般地，Horn 子句），则一定存在一个在同构意义下唯一的模型，从它到任一其他模型存在唯一的同态映射。这个模型称为该规约的初始模型。比如通常的自然数结构就是 NAT 的初始模型。采用初始模型作为代数规约的语义，就称为初始模型方法。

模块化的程序构造方式要求在已有规约的基础上引入新的类别和运算，即扩充式规约，或以满足某种要求的一类规约为参数，来定义新的规约，即带参规约。对于这些构造式的规约，除了初始模型语义外，还有终结模型或生成模型等不同的语义方法。

下面的例子给出了自然数上的“先进后出”栈的代数规约，这是一个扩充式规约。

例　自然数栈的代数规约

STACK_OF_NAT = **base** NAT

sorts nat　stk

opns EMPTY：　→stk

PUSH：　nat stk→stk

POP：　stk→stk

TOP：　stk→nat

ERROR：　→nat

eqns POP(PUSH(x,s)) = s

POP(EMPTY) = EMPTY

TOP(PUSH(x,s)) = x

TOP(EMPTY) = ERROR

代数规约用公理刻画程序的行为，抽象程度比较高。如何从给定的代数规约出发，得到能在计算机上执行的程序？这就是代数规约的实现问题。传统文献中关于这个问题的讨论多数局限于以低层规约实现高层规约，核心是实现的正确性。近年来人们开始探讨用传统高级语言中的模块实现代数规约，以及代数规约与前后断言规约的衔接问题。

如果将等式看成从左到右的重写规则，即可用其右边项替换左边项，一代数规约就成为项重写系统，从而可在计算机上直接执行，称为可执行的规约。

参考文献

1. Goguen J A, et al. An initial algebra approach to the specification, correctness, and implementation of abstract data types. In: Yeh R, ed. Current Trends in Programming Methodology, Vol. 4. Data Structuring. New York: Prentice-Hall, 1978
2. Ehrig H, Mahr B. Fundamentals of algebraic specification 1. Berlin: Springer-Verlag, 1985
3. Ehrig H, Mahr B. Fundamentals of algebraic specification 2. Berlin: Springer-Verlag, 1990

（林惠民）

daishuxue

代数学（algebra）　研究数字和文字系统运算理论及其结构的数学分支。

代数学的历史悠久，它随着人类生产技术的进步、科学和数学本身的需要而产生和发展。在这个过程中，代数学的研究对象和研究方法发生了重大的变化。代数学可分为初等代数学和抽象代数学两部分。初等代数学是更古老的算术的推广和发展，是研究实数或复数和以它们为系数的多项式的代数运算（加法、减法、乘法、除法、乘方和开方等）的理论和方法，其研究方法是高度计算性的，其中心问题是代数方程和代数方程组的解的求法及其分布的研

究。而抽象代数学则是在初等代数学的基础上产生和发展起来的,研究文字、向量、矩阵以及更一般的其他抽象对象的代数运算系统。

代数学的起源可以追溯到古巴比伦的时代,当时的人们发展出了较之前更进步的算术系统,使其能用字母和符号表示数量的方法来做计算,并能够列出含有未知数的方程并求解。代数学的西文名称 algebra 来源于9世纪阿拉伯数学家花拉子米的重要著作的名称。该著作名为"ilm al-jabr wa'l muqabalah",原意是"还原与对消的科学"。这本书传到欧洲后,简译为 algebra。代数学发展成为一门独立的数学分支应归功于中世纪的阿拉伯人。阿拉伯数学家系统地研究了二次方程的解法,确定了解方程求未知量是代数学的基本特征,建立了解方程的变形法则,还特别创造了三次方程的几何解法。在19世纪,代数学发生了革命性的变革。首先是挪威数学家阿贝尔证明了五次以上的一般代数方程不可能用根式求解。紧接着,法国数学家伽罗瓦对于高次方程是否能用根式求解问题给出更彻底的解答。他引进了置换群的正规子群、数域的扩域、群的同构等概念,证明了由方程的根的某些置换所构成的群(即伽罗瓦群)的可解性是方程根式可解的充分必要条件。伽罗瓦的工作最终导致抽象代数学的产生。

代数系统的研究成果在计算机科学和软件工程领域具有广泛的应用。

抽象数据类型的代数语义 数据类型和抽象数据类型均可视为一个代数系统,可以用一组代数公理来刻划数据类型操作的行为。因此,可以用代数公理方法定义抽象数据类型。代数学中用公理系统来研究代数结构时,不涉及其代数运算所作用的集合元素。与此类似,抽象数据类型的代数公理方法能够用与表示无关的操作的抽象性质来定义新的类型,将类型的对象作为抽象的元素,避开了新类型的对象的表示问题,从而避开了新类型尚未定义的问题。故代数公理方法的思想更符合抽象数据类型隐蔽表示细节和操作实现细节的"抽象"思想。

抽象数据类型及其实现的正确性 给出了抽象数据类型的实现,我们需要证明实现的正确性,即证明实现程序的语义所定义的代数与规范描述的语义代数是同构的。用高级程序设计语言建立抽象数据类型的实现过程在软件工程领域称为数据求精或模型转换。在代数学领域,求精和转换均可抽象为映射,因此可以用代数学中证明两个代数系统同构的方法证明抽象数据类型所有求精结果或不同模型的同构,进而得到最终实现的正确性证明。此外,对于同一数据类型的不同实现,如抽象堆栈可以用数组实现,也可以用链接表实现。可以用证明两个代数系统同构的方法,证明这两种实现方法功能上的等价性。

代数学在算法设计中的应用 对于给定的带权有向图,存在所有顶点对之间的可达性、最短路径和最大容量等多种实际问题和相关算法,这些算法分别由多位知名计算机科学家发明。可以将这类问题的共同特性抽象成一个代数结构闭半环。基于闭半环设计一个解带权有向图路径问题的泛型算法,算法中的数据和运算都是满足闭半环特性的抽象数据和抽象二目运算。然后将最短路径等实际问题抽象成具体的闭半环,用闭半环中的各个实际参数去替换泛型算法中的泛型参数(实例化),得到实际的路径算法。很显然,使用这样的方法可以大幅度提高算法设计的效率和可靠性。

参考文献

1. Stanat D F, Mcallister D F. Discrete mathematics in computer science. Englewood Cliffs, NJ: Prentice-Hall Inc., 1977

2. 朱一清. 离散数学. 北京:电子工业出版社,1996

(薛锦云)

daishu yuyi

代数语义(algebraic semantics) 用代数结构描述的计算机语言的**语义**。它把计算机语言的语义定义为满足某种公理体系的抽象代数结构,并利用这种代数结构来研究用该语言编写的程序的模型论性质。

代数语义始于对抽象数据类型的研究。数据类型是计算机语言中的重要组成部分,但在20世纪60年代中期以前一直缺少科学的定义。它被认为仅仅是一些数据的集合,这种观点不能反映数据类型的内在数学特性,因而不能用来检验程序的正确性。1967年问世的 SIMULA 67 语言第一次提出类型的概念,把数据和被允许施行于这些数据之上的运算结合为一个统一体,它是现代抽象数据类型的开端,但当时未引起足够重视。20世纪70年代初,软件危机促使人们去研究编写和验证正确的程序的理论和技术。在当时出现的一些新语言中,进一步把数据类型的特性与它的具体表示及实现方式分开来,提高了它的抽象程度。一个数据类型还可以规定对具有此类型的数据结构的表示和实现细节的屏

蔽方式,包括对从数据结构外部访问其内部和从内部访问其外部的限制,由此出现了完整的抽象数据类型的概念。

基于抽象数据类型的思想,用代数结构描述数据类型的语法(包括类子和类子间的运算结构)称为基调,再用一组公理描述上述运算的推导规则。基调加上公理就成为代数语义学意义下的抽象数据类型,满足这组公理的一个模型即是该抽象数据类型的一种代数语义,称为Σ代数,其中Σ代表基调,因此又称基调代数。当一个计算机程序被看成是抽象数据类型时,该抽象数据类型的代数语义就是此计算机程序的代数语义。

有些人(如 C. A. R. Hoare)是在研究公理语义的背景下讨论抽象数据类型的,他们关心的只是某段程序的执行是否满足某组前后断言,其重点在抽象数据类型的证明论性质。这个方法不能说明在满足公理组的诸模型中,哪些是设计者想要的,哪些是设计者不想要或希望排除的,这些模型之间的关系又如何,等等。从 20 世纪 70 年代开始,Goguen、ADJ 小组和 Guttag 等人提出了以抽象数据类型的模型集合及其性质为研究对象,采用模型论和范畴论方法给出了程序的代数语义理论。这种理论在 20 世纪 80 年代又得到了很大的发展。

公理集的性质　稳健性和完备性是一组对偶概念。已知在齐性代数(只有 1 个类子)中一定有一个完备且健康的证明系统。可是对于非齐性代数(类子数大于 1),Goguen 和 Meseguer 发现了一个具有不健康公理系统的抽象数据类型。这表明,应该引进一组元公理,作为抽象数据类型中公理集必须满足的条件,才能保证得到所需的模型。

为了使抽象数据类型具有模块化和可重用性质,一般都首先设计较小的抽象数据类型,然后逐步扩充,形成从小到大的抽象数据类型体系,扩充时遇到的问题是:小类型中原有的性质到大类型中会不会发生变化?这包括:小类型中原来不同的两个元素会不会由于大类型中新增加了一些等式公理而变成相等?小类型中原来属于某个类子的元素集会不会由于大类型中新增加了一些函数而膨胀起来?(例如,如果小类型中只有整数加法,大类型增加了整数乘法,但没有说明 $2\times4=8$,因此得到一个新的整数 2×4)。没有前一个问题的抽象数据类型称为是层次一致的,没有后一个问题的抽象数据类型称为是充分完备的。已知充分完备性是不可判定的,其他可判定性问题未完全解决。

模型集的结构　在同一抽象数据类型的两个Σ代数之间可以存在Σ同态关系。把Σ同态关系看成范畴论中的射,则同一抽象数据类型的所有Σ代数构成一个范畴。在某些条件下,例如当所有的公理都是Σ等式时,存在着在Σ同构意义下唯一的初始代数(初始模型)和终结代数(终结模型),采用初始模型作为语义的方法称为初始语义方法。如果只考虑有限生成模型,则当所有公理都是Σ等式时,同一抽象数据类型的全体有限生成Σ代数相对于该公理集取商得到一组模型,它们构成一个完全格。其中的最大元素和最小元素分别就是该抽象数据类型的初始模型和终结模型。

在一般情况下,终结模型是平凡的,即对每个类子只有一个代数元素与之相对应。有两种办法可以得到非平凡的终结模型。第一种办法是只考虑全体Σ代数的一个子集,在这个子集中寻找终结代数。第二种办法是在基层数据类型上建立扩充数据类型,然后在扩充数据类型中寻找终结模型。Wand 证明了,如果基层数据类型 B 相当“良好”地扩充为数据类型 E,使得这个扩充是层次一致和充分完备的,则在 E 的所有有限生成模型中必定存在一个终结模型 M,且 M 以 B 的初始模型为基子模型。Kamin 不需要层次一致和充分完备的条件,对任意的扩充数据类型 E,他根据Σ同态的概念,提出了一种行为等价的判定法则,把 E 中的全体有限生成模型按其是否行为等价分成许多等价类,并证明了每个等价类都有终结模型。当扩充数据类型满足层次一致和充分完备的条件时,只有一个行为等价类,于是回到 Wand 的结果。这种方法也适用于带参数的抽象数据类型。

程序语言的代数语义　在形式上,代数语义都是以抽象数据类型为背景研究的。如果能把一个计算机程序看成某种抽象数据类型,则也就有了该程序的代数语义。由于一个程序可能对某些输入是无定义的,因此要把上述抽象数据类型扩充为偏抽象数据类型,相应的Σ代数称为偏Σ代数。这里一个项 t 可以是有定义或无定义的,也可以是在某些模型中有定义,而在另一些模型中无定义(参见**Σ(基调)代数**)。计算机程序的另一个特点是两个不同程序可能会有相同的执行效果,这意味着两个不同的模型可能实际上是同一个。对扩充类型来说,它体现在如下定义上:项 t_1 和 t_2 称为强等价,若在所有模型中都有 $t_1=t_2$;它们称为弱等价,若把它们写成基层类型的项时,在所有模型中都等价。计算机

程序的代数语义性质可分为三个层次。第一层：对只含变量、常量、加、减、乘和布尔运算的表达式语言，存在充分完备和层次一致的抽象数据类型。它是多模型的，所有模型构成一个完全格。它各项的强等价性和弱等价性都是可判定的。第二层：对动态结构和静态结构一致的命令式语言，如果只含赋值和条件语句，则它只比第一层少一个弱等价可判定性。含静态循环（循环次数在进入循环前确定）时也是这样。若还含动态循环，则相应的抽象数据类型是层次一致的，但非充分完备，然而是弱充分完备。它是多模型的，但不构成完全格。第三层：对动态结构和静态结构不一致的命令式语言（例如含goto语句），可以建立相应的抽象数据类型，使第二层的性质在此还成立。它有一个弱初始模型和弱终结模型。它的全体极小模型构成一个完全格。

参考文献

1. 陆汝钤. 计算系统的形式语义. 北京：清华大学出版社，2017

2. Bergstra J A，Heering J，Klint P. Algebraic specification. New York：ACM Press，1989

（陆汝钤）

danhuilu shuzi kongzhiqi

单回路数字控制器（single loop digital controller） 一种以**微处理器**为计算与控制核心，配以相应软件，在外观及使用上类似常规模拟调节器的数字控制器。它一般可接收多个输入信号，但只输出一个模拟量信号（有的型号可以输出多个信号），构成单回路直接数字控制。它可以由用户编制程序，组成各种调节规律，所以又称为“单回路可编程控制器”。

单回路控制器一般由微处理器、过程输入输出通道、正面板、侧面板、供电电源、数字通信系统等硬件部分和监控系统、基本算式编程系统等软件部分组成。

单回路控制器将控制中常用的比例积分微分（PID）、超前与滞后（E/L）、四则运算、开方等几十种算式写入**只读存储器**中，称为“软件功能模块”。将这些功能模块根据用户的需要，按某种规律“连接”起来组成控制方案的过程，称为控制器的编程。编程工作通过专用的编程器或所附的编程器进行，方法有“在线”编程和“离线”编程两种。“在线”编程要求控制器中的随机存取存储器RAM有较大的容量，并有可靠的掉电保护装置。

图1为单回路控制器的内部输入输出关系图。由此图可以看出，算式的处理部分可由用户从标准算法库中任意选择并组态而成。

尽管单回路控制器的型号各不相同，但它们一般都具有以下共同特点：

（1）数字量和模拟量显示混合使用，输入和输出信号采用国际统一标准的模拟信号4～20 mA DC及1～5 V DC。

（2）外形结构、安装方式、正面操作面板的设置、操作及显示方式都与模拟调节器相似，操作

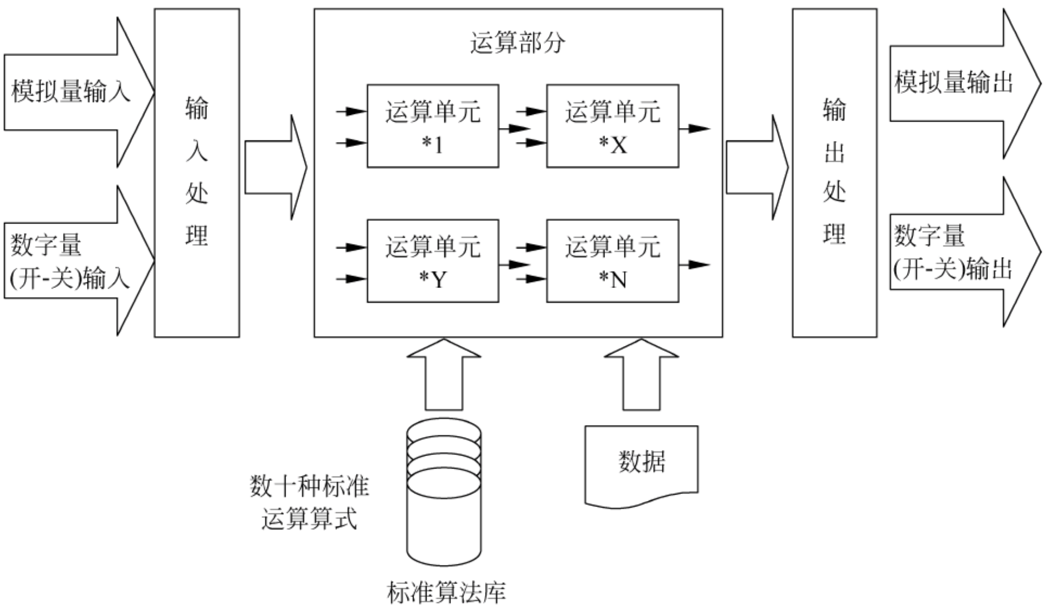


图1 单回路可编程控制器内部框图

简单。

(3) 用户程序编制采用“面向过程语言”,使用上类似袖珍计算器的编程,容易掌握。

(4) 控制器外部采用硬接线,与模拟调节器兼容;内部功能模块通过软件连接。控制器中配有数十种常用的控制算式和操作功能,用户可根据需要,按照系统的控制方案及算式,从中任意选择和组态。

(5) 具有数据通信功能。既能单独使用,也可与其他单回路控制器或数字仪表、CRT 操作站、上位计算机进行信息交换,构成不同规模的计算机控制系统。

(6) 具有自诊断功能。能对仪表的各个功能模块和软件进行在线检查,发现异常立即显示诊断代码,指出故障部位并随时报警。

(7) 在仪表的软硬件开发上采用了后备操作、后备电源、无扰动切换、故障自动切换与隔离及冗余措施等可靠性技术。

单回路控制器并不仅以取代以往的模拟调节器为目的,而是一台微型的工业控制专用计算机。现已被广泛地用于冶金、化工、石油、电力等工业自动化中。

随着计算机、数字通信、**人工智能**、自动控制等技术的迅速发展,当前的单回路控制器正走向高度的智能化,并与现场总线等技术结合在一起,以满足不同工业过程的不同控制要求,而新增的彩色 LCD 显示与增强的网络功能使新一代单回路控制器更加便于使用,功能更强。

参考文献

1. 周春晖. 过程控制工程手册. 北京:化学工业出版社,1993

2. 黄祯地,等. 过程控制仪表. 杭州:浙江大学出版社,1988

3. 王锦标. 计算机控制系统. 2 版. 北京:清华大学出版社,2008

(王慧)

danpian jisuanji

单片计算机(single-chip computer)　将**中央处理器**、存储器和输入输出接口集成在一个芯片上的**微型计算机**,简称单片机。由于单片机主要应用于控制系统,所以又称**微控制器**(MCU)。20 世纪 70 年代中期在推出通用微处理器的同时,也出现了 4 位和 8 位单片机。如 National Semiconductor 公司的 4 位单片机 COP 400、Intel 公司的 8 位单片机 8048 和 Fairchild 公司的 8 位单片机 F8。70 年代末 80 年代初,Intel 公司、Motorola 公司和 Zilog 公司分别推出了与当时 8 位微处理器功能相当的 8 位单片机,从而使单片机得到广泛应用。自 80 年代以来,不但推出了 16 位单片机,并使已广泛应用的 8 位单片机功能更完善,增强了片内 I/O 功能,片内存储器容量也进一步增大。进入 90 年代后,在进一步增强 8 位单片机功能的同时,推出了 32 位单片机,而 4 位单片机则逐渐淡出。进入 21 世纪之后,随着集成电路工艺的发展,单片机普遍采用了纳米级的工艺,大量 32 位单片机广泛地采用了 ARM 的 Contex-M 的核,使得 32 位单片机的价格能与 8 位单片机相当。

单片机一般都采用面向控制的系统结构和指令系统。其中有的采用了程序存储空间和数据存储空间相互独立的哈佛结构,如 Intel 公司的 8051 单片机;有的则采用类精简指令计算机(RISC)结构,如 Microchip 公司的 PIC 单片机和 ARM 架构的单片机。

单片机的片内存储器由片内随机存取存储器(RAM)和片内只读存储器(ROM)或 EPROM(参见**可擦编程只读存储器芯片**)、EEPROM(参见**电可擦编程只读存储器芯片**)和 FLASH 闪存组成。片内 RAM 较小,一般只有几 K 字节,用作通用寄存器、堆栈或数据存储器。片内程序存储器有片内掩膜 ROM、片内 EPROM、EEPROM 和 FLASH 等形式。根据应用批量的大小,可以选用上述几种形式之一;现在采用 FLASH 的形式变得越来越普遍。

单片机的多功能 I/O 结构是单片机的显著特点之一。较常见的通用 I/O 有定时计数器、并行 I/O 接口、串行 I/O 接口、USB 接口、CAN BUS 接口、A/D 转换器、D/A 转换器和 JTAG 仿真接口等;较常见的还有 LED 显示器、LCD 液晶显示器等的驱动电路、实时时钟、锁相电路、双音多频和声音合成电路等、触屏功能、无线传感器网络的 RF 电路和各种传感器等;有的单片机还具有浮点部件和 DSP 处理功能,如内核采用 Contex-M4 的单片机。若单片机要外接 I/O 及存储器,可用串行的 I2C 总线、SPI 接口和 SBI 接口等来外扩。

单片机的应用大致上可以分为家用电器应用、一般控制应用、汽车电子和便携式多媒体产品应用。在家用电器应用中,较多采用价格低廉的 4 位和 8 位单片机;在一般控制应用中,较多采用一般的 8 位单片机;在汽车电子应用中,较多采用高可靠性 8 位或 16 位单片机;在便携式多媒体产品应用中,则较

多采用32位单片机。

单片机的进一步发展有以下几个方面：①低电压、低功耗，可以在1.2 V电压下运行，电流仅为μA级；②片内I/O功能进一步增强，尽可能把应用所需的功能都集成在单片机内；③高可靠性和强抗干扰性；④作为嵌入式系统中的处理机，其处理功能越来越强大。

参考文献

1. 陈章龙. 单片微机十日谈. 北京：电子工业出版社，1995

2. 洪志刚. 单片机应用系统设计. 北京：机械工业出版社，2011

（陈章龙）

daozhuang xinpian hanjie

倒装芯片焊接（flip chip bonding）　一种用金属凸焊点将集成电路硅片上的压焊点与基板上的布线层直接键合的封装技术。由于焊点都放置在硅片的工作面（顶部），具体连接时，硅片是倒扣在基板上的，故称倒装芯片焊接，简称倒装焊。倒装焊、引线键合（wire bonding，WB）和载带自动焊（tape automated bonding，TAB）是芯片封装互连通常采用的三种技术。相对于传统引线键合技术，倒装焊中的芯片是一种倒扣形式，因此倒装焊技术又称倒扣芯片技术。芯片级倒装焊系统称倒装焊封装（flip chip in packge，FCIP），而在印制电路板上完成的倒装焊系统称为基板倒装焊（flip chip on board，FCOB）。

早在1964年倒装焊便设计使用在IBM公司的360系统的固态逻辑技术（SLT）混合集成晶体管器件上。倒装焊技术采用焊球作为互连媒介，将焊球直接制作在芯片上，在封装衬底上制作出相应的焊接图形（焊盘），芯片工作面向下将焊球与衬底焊盘对准后，所有焊点的焊接可以一次完成，省略了电连接引线的过渡，倒装焊焊球直接完成芯片和封装间的电连接，实现了芯片和封装间最短的电连接通路，具有卓越的电气性能。另外，芯片电极焊接点除边缘分布外，还可以设计成阵列分布，降低了封装密度的限制；倒装凸点等制备基本以圆片、芯片为单位，较单根引线为单位的引线键合互连来讲，生产效率高，降低了批量封装的成本。随着集成电路工艺进入深亚微米时代，倒装焊在封装体积及密度、电气性能、可靠性和生产效率等方面具有的明显优势，使其成为最有前途的芯片级互连技术。

倒装焊技术中关键工艺有多层金属膜（under bump metallurgy，UBM）制备、凸点制备、芯片倒装技术和底部填充技术。UBM是在芯片上的焊盘与凸焊点之间的一层金属化层，目的是使芯片与基板互连工艺更容易实现、互连可靠性更高。芯片上凸点的制备是倒装焊接的基础，在脆弱的芯片上制作一致性很好的凸点需要特殊的技术和工艺保证。芯片倒装技术主要有熔焊、热压焊、热声焊、胶粘连接等，这些技术中对准精度、间隙控制、压力和温度控制等具有较大难度。底部填充技术能够减少硅芯片和基板间热膨胀失配造成的影响，并能有效地缓冲机械冲击，填充材料和填充工艺是底部填充技术重点关心的问题。

参考文献

1. 中国电子学会生产技术学分会丛书编委会. 微电子封装技术——电子封装技术丛书. 合肥：中国科学技术大学出版社，2003

2. John H Lau，Shi Wei Ricky Lee. 芯片尺寸封装：设计、材料、工艺、可靠性及应用. 贾松良，等译. 北京：清华大学出版社，2003

（尚利宏）

dengzhimian chouqu jishu

等值面抽取技术（iso-surfaces extraction technique）　一种显式地构造数据场中的轮廓表面的绘制技术。等值面指空间数据场中的一个**曲面**，该曲面上各点的数据值均等于某一给定值V。等值面抽取技术被广泛应用于科学及工程计算数据场的建模。

早期的轮廓跟踪法在每个二维切片（slice）上提取二维等值轮廓线，继而连接属于同一对象的相邻切片上的等值轮廓线，得到三维等值面的三角面片集合。但当各切片上存在多个闭合的等值轮廓线时，相邻切片上的轮廓线可能差别很大，使得匹配十分困难，在处理复杂结构时则可能产生大量连接错误。

等值面提取方法中最经典的是由W. E. Lorenson和H. E. Cline于1987年提出的移动立方体法（marching cubes）。这一方法首先假定数据场的采样点均匀地分布在由立方体组成的三维网络的顶点上。且数据场沿立方体棱边线性变化。基于上述假设，可求出等值面与立方体棱边的交点，然后将它们按一定规则连接起来，得到近似表示等值面的一系列多边形或三角形。根据立方体顶点数值可能的分布，可分为256种情况。通过对称性简化，最终合并为15种处理情况（如图1）。该算法实现容易，现已得到广泛的应用。算法的缺点是：①产生大量散乱

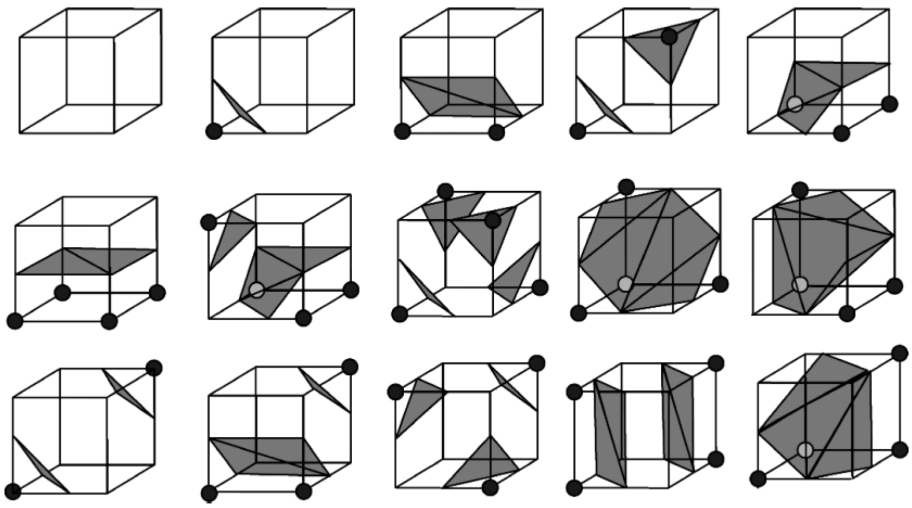


图 1　移动立方体法中等值面和立方体相交的 15 种不同的情况

的三角面片；②对某些层间距离较大的医学图像数据会产生"台阶"样的中间层；③构造等值面时存在二义性，对等值面显示速度、绘制效果及后继简化等影响很大。其中，产生歧义的情况，可以通过渐近线决策(asymptotic decider)法等解决。

当三维体数据的分辨率很高时，移动立方体法生成的三角面片通常很小，其投影甚至小于屏幕上一个像素点的面积。为避免插值计算三角面片，可采用剖分立方体(dividing cubes)方法。这种方法逐个扫描每个体元，当体元的 8 个顶点跨越等值面阈值时，将该体元递归细分，直到其投影面积小于显示图像像素，将其作为一个点投影显示到输出图像上。

对于不规则体数据，可将输入数据剖分为四面体集合，再通过移动四面体(marching tetrahedra，MT)的方法获得等值面。这一方法和移动立方体方法类似，但顶点的取值只有两种不同的情况，即只有一个顶点的值大于等值面阈值(和三个顶点的值大于等值面阈值情况对称)和两个顶点的值大于等值面阈值的情况，对应于生成等值面上的一个或两个三角形。由于四面体是最简单的多面体，其他任何类型的多面体都可以剖分解析为四面体，因此该方法可以适用于各类体数据。此外，移动四面体法不存在二义性问题。

对于大规模的体数据，已提出了各种加速算法以高效地提取等值面。在提高移动立方体算法的效率方面有基于八叉树的等值面提取方法，即在八叉树的每个结点上存储其所有子结点的最大值与最小值。遍历八叉树时，根据子结点的取值范围，直接跳过不可能含有等值面的部分。与基于几何空间分解的加速方法不同，基于范围查询(range-query)的算法通过建立跨度空间(span Space)，在数值空间上对体数据进行分解，可同时应用于规则或者不规则数据。此外，通过视点相关的方法，只考虑可见部分的等值面提取也能提高效率。

参考文献

1. Lorensen W E, Cline H E. Marching cubes: a high Resolution 3D surface construction algorithm. Computer Graphics, 1987, 21(4): 163-169

2. Hansen C D, Johnson C R (eds.) The visualization handbook. Elsevier, 2005

(袁晓如)

digonghao sheji

低功耗设计(low power design)　以降低集成电路功耗为目标的芯片设计方法和流程。随着半导体工艺的飞速发展和芯片工作频率的提高，芯片的功耗迅速增加，而功耗增加又将导致芯片发热量的增大和可靠性的下降。因此，功耗已经成为深亚微米集成电路设计中与性能、面积同等重要的一个考虑因素。进行低功耗设计可以节省能源、降低成本、增强系统的稳定性及提高系统的性能。低功耗设计涉及集成电路设计的各个层次，包括系统级、体系结构级、寄存器传输级、门级、电路级和工艺级。

CMOS 电路的功耗主要由 3 部分组成：电路工作时电容充放电导致的功耗、短路电流引起的功耗(合称为动态功耗)，和结反偏时漏电流引起的功耗(也称为静态功耗)。即 $P_{total} = P_{turn} + P_{short} + P_{leakage}$，其中：

P_{turn}　即跳变功耗，是器件在工作过程中对电

容充放电形成的。$P_{turn} = \alpha * C_L * V_{dd}^2 * F$,其中 F 是系统频率,α 是跳变因子,即整个电路的平均翻转比例,C_L是门电路的总电容,V_{dd}是供电电压。

P_{short} 短路功耗,也叫直通功耗,是器件在工作时由电源到地形成的通路造成的(电路稳定时,P 管和 N 管中总有一个是截止的,但在翻转的瞬间,P 管和 N 管会同时导通,形成短路电流 I_{short})。$P_{short} = \tau * \alpha * V_{dd} * I_{short}$,其中 τ 是电压信号上升/下降的时间。

$P_{leakage}$ 漏电流功耗,是由扩散区和衬底之间的反向偏置漏电流 $I_{leakage}$ 产生的。$P_{leakage} = V_{dd} * I_{leakage}$。

集成电路的低功耗设计可以体现在设计的各个层次当中。目前,在集成电路设计中采用的低功耗方法主要有:在工艺级,通过降低电源电压,减小晶体管尺寸,增加金属层数及采用其他特殊工艺等技术实现;在电路级,主要针对跳变功耗,涉及电源电压、物理电容和开关频率几个方面,最常用的是通过门控时钟或门控电源,关闭当前不使用的部分电路的时钟(以减少不必要的跳变)或电源(以减少漏电);在门级,通过门尺寸优化和门级多阈值电压技术实现;在寄存器传输级,减少寄存器不希望的跳变;在体系结构级,通过并行结构、流水线结构技术实现。在系统级,通过动态电源电压管理、动态阈值调节和休眠模式等技术实现低功耗。在实际设计中,往往是在多个层次利用多种技术配合以实现整体功耗的降低。

低功耗设计是平衡的艺术。通过降低供电电压和时钟频率等都可以降低功耗,但系统的性能也会受到影响,因此,如何在系统的性能和功耗之间做权衡是低功耗设计必须考虑的内容。通常的解决办法是,利用硬件实时捕获负载的性能需求,再反馈给软件,通过软件动态控制和调整硬件的参数,追求以最低的功耗代价满足性能的需求,或者在一定的功耗代价下,获得更高的任务吞吐率。此外,通过流水线或并行结构的方法降低功耗,也必然会增加芯片的面积。所以,具体的芯片设计时,在哪些设计层次、采用何种方法降低功耗,需要根据系统各个方面(性能、面积、功耗、价格等因素)的设计需求,综合考虑权衡而定。

参考文献

1. Mudge T. Power: A first-class architectural design constraint. IEEE Transaction on Computer. 2001, 34(4)

2. 陈春章,等. 数字集成电路物理设计. 北京:科学出版社. 2009 (范东睿)

diji yuyan

低级语言(low level language) 与特定计算机体系结构密切相关的**程序设计语言**。它包括字位码、**机器语言**和**汇编语言**。字位码是计算机唯一可直接理解的语言,但由于它是一连串的字位,复杂、烦琐、冗长、几乎无人直接使用。机器语言是表示成数码形式的机器基本指令集,或者是操作码经过符号化的基本指令集。汇编语言是机器语言中地址部分符号化的结果,其指令与计算机指令通常是一一对应的,或进一步包括宏构造。用它们书写的程序不必经过翻译或只经过简单的翻译后就可以在计算机上执行。

低级语言的特点是与特定的机器有关,功效高,但使用复杂、烦琐、费时、易出差错,程序员用数码形式或符号化的基本指令构造复杂的程序会由于涉及太多的琐碎问题而工作量太大且易出错。通常使用低级语言进行程序设计目的是用它们可以写出执行速度更快且占用更小内存的程序。

参考文献

Sammet J E. Programming language: history and fundamentals. Englewood Cliffs, NJ: Prentice Hall, 1969 (郑国梁)

dili xinxi xitong

地理信息系统(geographic information system, GIS) 支持地理空间数据的采集、管理、处理、分析、建模和显示的一种**应用软件系统**。主要用于地理信息的查询,空间问题的规划、管理和决策。地理空间数据包括与空间要素几何特性有关的空间数据以及提供非空间信息的属性数据。GIS 以地理空间数据作为主要操作对象,这是它区别于其他类型信息系统的主要特征。目前,地理信息系统已经发展成为一门融合了计算机科学、地理科学、测绘科学、统计分析的综合性学科。

地理信息系统萌发于 20 世纪 60 年代初。1960 年,加拿大 R. F. Tomlinson 提出"把地图变成数字形式,以便于计算机处理和分析"的思想。1965 年,W. L. Garrison 正式提出地理信息系统这一术语,1971 年加拿大土地管理局建成世界上第一个用于开展土地资源调查的加拿大地理信息系统 CGIS。70 年代和 80 年代,随着计算机技术,特别是网络技术的发展,GIS 在自然资源和环境数据处理中得到应用,并出现了 ARC/INFO、GENAMAP、MICROSTATION 和

SYSTEM9 等 GIS 工具软件，其应用从解决基础设施的规划，转向更复杂的区域开发和全球性的问题，如全球可居住区的评价、厄尔尼诺现象及酸雨、核扩散等对世界环境潜在的影响等。90 年代后期，网络服务技术的出现和发展，为实现基于 Web Service 的信息共享与空间数据互操作奠定了基础。21 世纪以来，GIS 在资源、环境、交通、电信、能源、农业、林业、水利、国防、公安、航空航天以及数字省区、数字国土和数字海洋等领域得到广泛应用，并开创了网格 GIS 和移动 GIS 发展的新格局，正朝着地理信息服务的方向不断延伸和发展。现在，GIS 经过 50 年的发展历程，涉及的学科和产业已发展为颇具生命力的地理信息科学和地理信息产业。

在地理学理论依托下，GIS 主要研究内容有：空间实体的表达和建模，空间数据的获取和管理，空间数据的处理和转换，空间信息的查询和分析，GIS 应用模型的构建，GIS 的系统设计与评价，地理信息标准化研究，空间信息基础设施建设，GIS 产品的可视化方法以及地学信息传输机理的不确定性（多解）与可预见性（多维）的研究等。

地理空间的特征实体包括点、线、面、曲面和体等多种类型，它们具有空间、属性和时态等特征，反映这些特征的数据称为空间数据。因此，空间数据的本质是每一个空间实体都按统一的地理坐标或格网进行编码，实现对其定位、定性、定量和拓扑关系的描述，提供作为 GIS 处理和操作的主要内容。分析空间实体的信息特征，研究空间实体的数据建模方法，如何以结构化和半结构化数据表达实体的特征以及空间数据分类、编码、组织和管理等，是 GIS 的基础性研究课题。

GIS 的基本功能见图 1 所示的五大模块或五大子系统。数据输入与转换子系统将现有地图、外业观测成果、航空相片，遥感数据、文本资料等转换成数字形式，并将数据归化后进入空间数据库中。图形及属性编辑子系统涉及编辑修改原始输入数据的错误，进行图形变换和装饰，建立拓扑关系和图形接边，输入属性数据，实现图形数据与属性数据的连接等。空间数据库管理子系统涉及地理空间实体的坐标数据、拓扑数据和属性数据的组织和管理。空间查询与分析子系统通过空间查询语言和空间分析算法，使之支持位置查询、属性查询和拓扑查询以及数字地形模型分析、空间叠合分析、缓冲区分析、网络分析、集合分析和决策分析等。制图与产品输出子系统将系统处理的结果数据最终加工成各类地图图

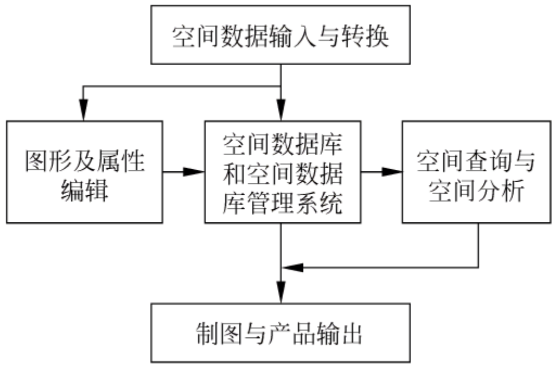


图 1　GIS 基本功能模块

形输出，也包括图像、图表和文字说明等形式的输出。

GIS 涉及的学科除计算机科学外，还有地理学、测绘学和数学等，明显地体现出多学科交叉的特征。计算机科学为 GIS 提供了系统构成、软件设计、数据组织、数据管理、图形生成和系统建立等的理论和方法，计算机领域的许多技术如**数据库系统**、**面向对象程序设计**、**多媒体技术**、三维技术、**虚拟现实**、图形图像处理和**人工智能**等，都与 GIS 的发展有着密切的联系。地理学广泛涉及居住的地球和地理空间，这与 GIS 的研究对象是一致的，地理学中的空间分析正是 GIS 空间分析的核心。测绘学及其分支学科，如大地测量学、摄影测量学、地图学等，不但为 GIS 提供各种不同比例尺和精度的定位数据，它们中的误差理论、地图投影理论、图形学理论、许多相关的算法等可直接用于 GIS 空间数据的变换和处理以及 GIS 产品的开发和设计等。数学的许多分支学科，包括几何学、统计学、运筹学、数学形态学、图形代数、拓扑学、图论、分形分维理论等，已经广泛应用于 GIS 空间数据的分析、应用模型的构建和空间实体的表达。

当前，随着数字地球和智慧城市等信息化浪潮的兴起，GIS 正向集成化、产业化和服务化的发展方向迈进。

参考文献

黄杏元，马劲松．地理信息系统概论．3 版．北京：高等教育出版社，2008

（黄杏元）

digui fa

递归法（recursive approach）　一种一般的算法设计方法。用递归法求解给定问题时，设计的算法一次或多次递归地调用自身以处理密切相关的子

问题。

递归法之所以重要是因为递归计算基于已知值计算未知值,从有限个值计算出无限个值的过程在某种意义上反映了计算的本质。并且理论上已经证明图灵可计算函数类 = μ 递归函数类,这意味着建立在图灵机模型上的现代计算机可计算的函数都是 μ 递归函数,因而可基于零函数、后继函数和投影函数,通过函数合成、原始递归和 μ 运算取极小把它们构造出来。进一步,并非一切可计算函数都能用显式定义,“迭代”是递归的特殊情况,递归定义往往十分简洁,其运行效率可借助程序变换和优化技术来提高。

函数的递归定义分两步:

(1) 显式给出若干初始值;

(2) 按某种序用已知值定义未知值。

第一步中需给出的初始值的数量由第二步来确定,即依赖于第二步定义未知值时需要多少已知值。例如,阶乘函数和斐波那契级数可分别递归定义为

$$\begin{cases} 0! = 1, & n = 0 \\ n! = n \times (n-1)!, & n > 0 \end{cases}$$

$$\begin{cases} F(0) = 0, & n = 0 \\ F(1) = 1, & n = 1 \\ F(n) = F(n-1) + F(n-2), & n > 1 \end{cases}$$

一般来说,若问题本身具有明显的递归特征,则可用递归算法。若栈的使用是不可避免的(如二叉树的中根遍历问题),则最好采用递归算法。因为在实现递归过程时,编译程序会自动建立并管理一个由栈块组成的调用堆栈。一个栈块中包含实参指针(或内容)、局部变量的值和返回地址。这样,实现调用语句时,在调用堆栈中建立一个新的栈块然后实现控制转移。实现返回语句时,首先处理结果表达式并取出返回地址,然后释放栈块,最后实现控制返回。在按“总时间 = 调用次数 * 一次调用的时间”来分析递归过程的时间复杂性时,由于一次调用中又含递归调用,所以这个等式实际上是一个递归方程。求解递归方程的一种思路是:根据递归方程反复展开后应用等比数列的求和公式求和。另一种思路是:借助生成函数来求解递归方程。已知 $F(n)$ 的递归定义,使用生成函数求 $F(n)$ 的解析(显式)定义的步骤如下:

(1) 构造 $\{F(n)\}n=0,1,\cdots$ 的生成函数 $G(t) = \sum_{i\geqslant 0} F(i)t^i$;

(2) 根据 $F(n)$ 的递归定义式的特征,借助加、减、乘、除、微分、积分等手段,使级数除有穷项外,其他项的系数全为0,从而得到关于 $G(t)$ 的有限方程,最后求出 $G(t)$ 的有限形式;

(3) 将 $G(t)$ 的有限形式重新展开成 t 的幂级数 $G(t) = \sum_{i\geqslant 0} H(i)t^i$,则 $F(i) = H(i), i = 0, 1, \cdots$

作为一种一般的算法设计方法,递归法可用于求解各种各样的问题(如:整数分划数问题)。

参考文献

1. Aho A V, Hopcroft J E, Ullman J D. The design and analysis of computer algorithms. Boston, MA: Addison-Wesley, 1974

2. Cormen T H, Leiserson C E, Rivest R L, et al. Introduction to algorithms. 3rd ed. Cambridge, MA: The MIT Press, 2009

3. Kleinberg J, Tardos E. Algorithm design. Boston, MA: Addison-Wesley, 2005 (殷建平)

digui hanshu

递归函数(recursive function) 一种以其早先的值来定义新值的可计算函数。

早在1202年,意大利数学家L. Fibonacci提出兔子繁殖问题,就开始了对递归的研究。假设兔子繁殖规则为:①每对成熟兔子每月繁殖一对后代;②兔子出生后第二个月成熟;③老兔子不死。如果1月底引入一对刚出生的兔子,问年底有多少对兔子?

用 FIB(N) 表示第 N 月底兔子对数,则

$$\mathrm{FIB}(N) = \begin{cases} 0, & \text{如果 } N=0 \\ 1, & \text{如果 } N=1 \\ \mathrm{FIB}(N-1) + \mathrm{FIB}(N-2), & \text{如果 } N \geqslant 2 \end{cases}$$

这个函数有一个突出的性质,即用早先的值来定义新值。这就是递归定义。

所谓***递归***,是将一较大问题归约到一个或多个子问题的求解过程,这些子问题在结构上与原问题相同,但子问题求解比原问题简单一些。全面系统地研究递归函数,应归功于K. Gödel,他在研究希尔伯特规划时,发明并使用了原始递归函数的形式系统。

在定义递归函数之前,先定义函数复合、递归、取极小三种运算:

复合 函数 $f(y_1, \cdots, y_k), g_1(x_1, \cdots, x_n), \cdots, g_k(x_1, \cdots, x_n)$ 的复合函数定义为

$$h(x_1,\cdots,x_n)=f(g_1(x_1,\cdots,x_n),\cdots,g_k(x_1,\cdots,x_n))$$

递归　函数$f(x_1,\cdots,x_n)$,$g(x_1,\cdots,x_n,y,z)$通过递归运算所得函数$h(x_1,\cdots,x_n,y)$由下列递归方程唯一确定:

$$\begin{cases} h(x_1,\cdots,x_n,0)=f(x_1,\cdots,x_n) \\ h(x_1,\cdots,x_n,y+1)=g(x_1,\cdots,x_n,y, h(x_1,\cdots,x_n,y)) \end{cases}$$

取极小　对函数$f(x_1,\cdots,x_n,y)$取极小,得

$$g(x_1,\cdots,x_n)=\mu y(f(x_1,\cdots,x_n,y)=0)$$

$$\begin{cases} =k, & \text{如果对所有 } z<k,\ f(x_1,\cdots,x_n,z)\text{有定义且不为0,而} f(x_1,\cdots,x_n,k)=0; \\ \text{无定义}, & \text{如果不存在这样的 } k \end{cases}$$

取极小算子通常称为μ算子。

部分递归函数类是含零函数0,后继函数$x+1$,投影函数$U_i^{(n)}$,并且在复合、递归、取极小运算下封闭的最小函数类。其中,投影函数$U_i^{(n)}(x_1,\cdots,x_n)=x_i$,$n\geqslant1$,$1\leqslant i\leqslant n$。

上述定义中,若不含取极小运算,则得到**原始递归函数**类。

递归函数类与图灵可计算函数类相同,只要计算机能容许任意多和任意大的整数,该类亦与由PASCAL或FORTRAN程序定义的整数函数类相同。无论是程序还是一般递归模式,大多只能得到部分函数,因为对某些初值计算可能不终止。

可计算函数属递归函数论的研究范畴,递归函数论是数学领域的一活跃分支。计算机科学的实践与递归函数论有相当密切的联系,递归的思想影响了程序设计语言的构造,也影响计算机系统的结构。例如,递归过程、递归数据结构、堆栈等都已成为计算机科学中的常用词汇了。

参考文献

1. Kleene S C. 元数学导论. 莫绍揆,译. 北京:科学出版社,1985

2. Cutland N. Computability, an introduction to recursive function theory. Cambridge, UK: Cambridge University Press, 1980

(陈火旺　贲可荣)

dianyun

点云(point clouds)　通过三维扫描仪获取的离散的采样点集。通常包含采样点的三维坐标、法向以及表面外观属性。点云可用来表示连续的三维模型外表面。由点云构成的曲面称为点集曲面,又称点模型。继以三角网格为几何模型的表达方式之后,点云成为几何模型又一种新的表达方式。以点云为研究对象的点云几何处理也成为**计算机图形学**中的一个新的研究领域。

点云表示几何模型的优点是:与传统的三角网格模型相比,不需要存储和维护庞大的拓扑连接信息;能表示几何模型丰富的几何细节;由于现代三维数字摄影技术和三维扫描系统的发展,其数据获取方便。缺点是,点云是由一些没有任何拓扑连接关系的离散点所构成,在点云上定义几何算子有难度;点云通常数据量大,如何快速有效地进行几何处理需要深入研究。

通常有两种获取点云数据的方式:第一,通过交互的曲面造型软件和算法构建;第二,采用三维扫描仪(如激光测距扫描,结构化的光学扫描仪等)对物理模型进行数字采样。随着三维几何模型越来越复杂,第一种方法由于耗时且难以构建现实世界中的物理模型,使第二种方法逐渐占据主导地位。如今各种低档和高档三维扫描仪的三维几何获取能力以及数据质量已足以满足实际应用的要求。

基于点云的数字几何处理是点云的主要研究内容。狭义的点模型数字几何处理的研究内容包括几何处理(processing)和几何造型(modeling)。对点云的几何处理算子有曲面重建、基于点云的绘制、曲面分析、滤波、重采样、分割、形状匹配、特征分析、特征提取等;而造型的研究内容有布尔求交、自由变形、表面着色、纹理绘制、雕刻等。广义的点云数字几何处理还包括几何修复、动画、压缩、编码、传输等。点云的数字几何处理是一个十分广泛的研究领域,学术界已在该领域开展了大量的工作,并且方兴未艾。

基于点云**数字几何处理**已获得很大进展,如下几何领域是点云有意义的研究方向:①点云模型的压缩、传输以及数字水印;②基于点云表示的自然景物建模;③基于点云的动画(爆炸、烟雾等效果)。

参考文献

Gross M, Pfister H P (eds.) Point-Based Graphics. San Francisco, CA: Morgan Kaufmann, 2007

(肖春霞)

dianci jianrongxing

电磁兼容性(electromagnetic compatibility, EMC)　电子设备(或系统)在共同的电磁环境中能一起执行各自功能的共存能力。该设备(或系统)不会由于受到处于同一电磁环境中的其他设备

(或系统)的电磁发射而导致或遭受不允许的降级,也不会使同一电磁环境中的其他设备(或系统)因受其电磁发射而导致或遭受不允许的降级。电磁环境包括设备(或系统)的辐射电磁环境和自然界中的电磁环境两大类。要求设备(或系统)在规定的电磁环境中按设计要求具有一定的抗干扰能力和不产生超过限度的电磁干扰,并能正常工作。

电磁干扰是人们早已发现的问题。1888 年赫兹用实验证明各种打火系统向空间发射电磁干扰,1934 年英国有关部门对一系列的干扰问题进行分析研究后发现,有 50% 的干扰是由电气设备引起的。为了加强对干扰问题的深入研究,国际电工委员会(IEC)成立了国际无线电干扰特别委员会,专门从事有关无线电干扰标准的研究和制定工作,其内容包括:保护无线电装置免受某些干扰的措施,干扰的测试方法及设备,干扰源产生干扰的允许值,电子设备的抗干扰度及其测试方法等。电磁干扰问题虽然由来已久,但电磁兼容这一新的学科却是近代才形成的。自 20 世纪 80 年代以来,世界各国在电磁兼容标准与规范、EMC 设计与测量、EMC 材料与工艺等方面进行了大量研究,使 EMC 的设计水平有了很大的提高。我国在 1966 年制定了第一个干扰标准:JB 851《船用电气设备工业无线电干扰端子电压测量方法和允许值》。20 世纪 80 年代以来,先后又制定了 30 余项有关 EMC 的国家标准、国家军用标准和行业标准。这对推动我国 EMC 技术的发展和提高电子产品的可靠性,都起到了积极的保证作用。目前已有高精度的电磁干扰及其敏感度的自动测试系统,设备内及设备之间 EMC 的计算机辅助分析程序,从而形成了一套较完整的 EMC 设计体系。

电磁兼容性研究的基本内容包括:电磁干扰特性及其传播方式,电磁兼容性设计技术,电磁兼容性频谱利用,电磁兼容性材料与工艺,电磁兼容性测试和模拟技术,电磁兼容性规范与标准等。进行电磁兼容性设计时,应明确设备(或系统)在多强的电磁干扰环境中能正常工作以及本系统干扰其他系统的允许值;了解设备(或系统)的干扰源、被干扰源、干扰耦合途径等;根据具体要求,采取相应措施抑制干扰源,消除干扰耦合途径,提高电路的抗干扰能力。

电磁干扰源可分为自然干扰和人为干扰两种形式。自然干扰主要是雷雨、闪电产生的天电噪声,太阳黑子爆炸活动及银河系辐射产生的宇宙噪声等。人为干扰是由机电或其他人工装置产生的电磁干扰,如各种信号发射机、振荡器、电动机、开关、继电器、机动车辆的点火系统、家用电器、高速逻辑电路、门电路、晶闸管逆变器、整流器、照明设备、信息处理设备以及核爆炸时的核电脉冲等。随着科学技术的发展,人为干扰已成为电磁干扰的主要来源。为保证通信、广播与电视等电子设备正常运行,国际无线电干扰特别委员会制定了工业、科学、医用和家用电器及电动工具等人为干扰源的干扰极限值。我国在相应的国家标准及军用标准中也作了规定。这些标准有 GB 4343《电动工具、家用电器和类似器具的无线电干扰特性测量方法和允许值》,GB 4824《工业、科学和医疗射频设备无线电干扰特性测量方法和允许值》,GJB 151《军用设备和分系统电磁发射和敏感度要求》等。

电磁干扰传播按其耦合途径可分为传导和辐射两种类型。沿电源线或信号线传输的电磁干扰称传导干扰,包括通过公共电源线直接传导,通过公共电源内阻耦合和经公共地线阻抗耦合而进入被干扰对象等几种形式。通过空间传播的电磁干扰称辐射干扰,干扰源中各种信号电路、电源电路导线在一定条件下可构成辐射天线,当干扰源外壳流过高频电流时,此外壳也将成为辐射天线向外辐射电磁能。

干扰源周围空间可分为近场和远场两种区域,距离小于 $\lambda/2\pi$(λ 为波长)的区域为近场区,反之为远场区。近场区内有电容耦合和电感耦合两种形式。远场区的电磁能量以电磁波的形式通过空间传播,作用到被干扰对象。

评定电磁兼容性能的指标是屏蔽效果。随着计算技术的发展,利用数值计算方法,可以对各种屏蔽结构的屏蔽效果进行定量分析和仿真模拟试验,其主要计算方法有基于传输和散射理论的屏蔽效果计算方法,基于并矢格林函数的孔缝耦合计算方法以及时域有限差分法等。

应用于电磁屏蔽的材料有衬垫材料、阻隔材料、屏蔽器件等。衬垫材料包括编制丝网、导电硅橡胶、金属丝纤维和网屏衬垫等,阻隔材料有屏蔽视窗、通风孔板、导电黏接剂等,屏蔽器件包括开关和转动部件的密封衬垫、箔带等。

提高电磁兼容性的具体措施有:屏蔽(电屏蔽、磁屏蔽和电磁屏蔽)、滤波、接地、限幅,恰当分配系统频率,正确选择连接电缆和布线方式,采用平衡差动电路、整形电路、积分电路和选通电路等技术。

参考文献

1. B E 凯瑟. 电磁兼容原理. 肖华庭,等译. 北京:电子工业出版社,1985

2. 顾希如. 电磁兼容原理规范和测试. 北京:国防工业出版社,1988 (赵惇殳)

dianke ca biancheng zhidu cunchuqi xinpian

电可擦编程只读存储器芯片(electrically erasable programmable read only memory chip) 一种存储内容可在电路系统中写入和擦除,而去电后所存数据不会丢失的(非易失性)半导体**只读存储器芯片**(简称 EEPROM 芯片)。它不需专门的紫外线擦除设备。虽然 EEPROM 芯片可像随机存取存储器(RAM)芯片那样在系统中随机读写,但写入周期时间长,为毫秒级,而且允许写入的次数也是有限的,一般不大于 10^5 或 10^6 次。因而不能作 RAM 芯片使用。

早在 1970 年就开始使用金属-氮化物-氧化物硅(MNOS)栅区制作 P 沟道单元 EEPROM 芯片,不久又制作出 N 沟道的。1984 年至 1985 年期间集成规模已发展到 64 kb,由于减少功耗和设计专用逻辑的需求,促成了 CMOS(互补金属-氧化物-半导体)工艺 EEPROM 芯片发展。以 64 kb EEPROM 芯片为例,其静态功耗由 NMOS 时的 200 mW 降到 CMOS 时的 5 mW。

类似于紫外线 EPROM(UV EPROM),EEPROM 的存储单元也用多晶硅浮栅上有无电荷来区分 0/1 状态。浮栅电荷来自编程时高电压下雪崩击穿时浮栅俘获的电子。EEPROM 不用紫外光擦除,将浮栅与衬底(沟道)和漏极间绝缘层厚度减到 10 nm 以下,在高电压下通过隧道效应,电子可从浮栅移除,实现电擦除。当高电压反向时隧道效应也可实现编程写入(移入电子)。EEPROM 的存储单元相当于由一个读出管和一个浮栅管组成。通过源极、栅极、漏极和衬底不同的电平组合来实现擦除、写入和读出,并且可实现按字节或页面擦除和编程。

初期 EEPROM 的擦除和写入的高电压由外部输入,因而需二种电源(芯片 CMOS 电路工作电源 V_{DD} 和擦除/编程所需的高压电源 V_{PP})。应用电荷泵技术实现片内升压后,可使用单一电源。但编程和擦除的时间加长。

EEPROM 按芯片信号接口可分为串行 EEPROM 和并行 EEPROM 芯片。串行 EEPROM 芯片有三种主要的总线接口和协议(串行数据的排列规则和时序)的芯片:I^2C、SPI 和 Microwire 总线的芯片。串行 EEPROM 芯片主要为满足低成本、低速度和小容量的市场需求,主要用于存储数字系统的初始配置信息。例如内存条上 256 字节的 I^2C 接口串行 EEPROM 用来存储内存条的 DRAM 芯片类型、组织结构、时序要求等信息,以便运行 BIOS 时,设置存储器控制器,保证内存条的正确工作。并行 EEPROM 芯片有与 EPROM/SRAM 相同的地址、数据和读/写/选片控制,因而容易与微机芯片相连,并可按字节或字进行擦除和写入,主要用于开机导引程序和 BIOS 的存储。为避免误写入,有软件数据保护流程算法,向若干规定地址写不同规定的数据,使芯片进入或退出数据保护状态。

快闪 EPROM 芯片(参见**快可擦编程只读存储器芯片**)实际上也是一种 EEPROM,其中的 NOR 闪存芯片读出速度快,但存储单元面积大,不适合大容量集成。其接口与上述并行 EEPROM 相同,主要用于数字系统的开机导引程序和基本输入输出系统(BIOS)的存储和中等容量的数据存储。NOR 闪存出现后,一般提 EEPROM 时,多指串行 EEPROM,而现在的 PC 已采用串行外设接口(SPI)的 NOR 闪存。

参考文献

1. Sharma A K. 先进半导体存储器——结构、设计与应用. 曾莹,等译. 北京:电子工业出版社,2005

2. 桑野雅彦. 存储器 IC 的应用技巧. 王庆,译. 北京:科学出版社,2006

3. http://www.samsung.com/Products/Semiconductor

(孙祖希)

dianlan chuanshu jiezhi

电缆传输介质(cable transmission media) 网络数据传输中发送方与接收方之间的物理通道,常用介质为铜导线。根据使用类型,电缆传输介质分为两种,即双绞线电缆和同轴电缆。

双绞线电缆(twisted pair cable) 由多对双绞线包裹在一个绝缘护套内构成的。双绞线(twisted pair)是一种由两根覆盖绝缘材料的金属导线按一定的规格互相绞合在一起制作而成的传输介质。双绞线是综合布线系统中最常用的一种介质,用于电话通信和以太网络。双绞线由 Alexander Graham Bell 在 1881 年发明,1900 年开始用于美国的电话线网络。

双绞线电缆分为非屏蔽双绞线(unshielded twisted pair, UTP)与屏蔽双绞线(shielded twisted pair, STP)两类,如图 1 和图 2 所示。屏蔽双绞线相对于非屏蔽双绞线的区别主要是屏蔽双绞线内双绞线对

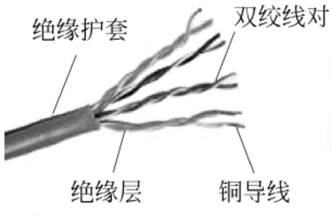


图 1 非屏蔽双绞线(UTP)

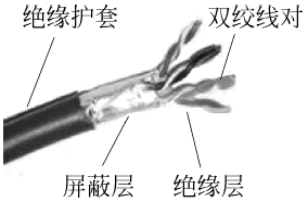


图 2 屏蔽双绞线(STP)

的外侧与外层绝缘护套之间有金属编织网或金属箔构成的屏蔽层。

非屏蔽双绞线电缆用于各种网络,由 4 对共八根外侧覆盖绝缘材料的铜导线组成,每一对中铜导线两两相互缠。屏蔽双绞线同样由 4 对共八根外侧覆盖绝缘材料的铜导线组成,每一对双绞线对外侧覆盖着一层金属箔,同时 4 对双绞线对与外层绝缘护套之间还覆盖有金属编织网或金属箔。金属编织网或金属箔既可减少辐射,也可阻止外部电磁干扰。屏蔽双绞线的价格比非屏蔽双绞线贵,施工安装时也要相对困难。

常用的双绞线类型及特性如表 1 所示。

表 1 双绞线类型及特性

类别	类型	带宽/MHz	主要应用
一类	UTP	0.4	电话
二类	UTP	1	终端通信
三类	UTP	16	以太网 10BASE-T、100BASE-T4
四类	UTP	20	令牌环网 16 Mb Token Ring 以太网 100BASE-T4
五类	UTP、STP	100	以太网 100BASE-T4、100BASE-TX、1000BASE-T
超五类	UTP、STP	100	以太网 100BASE-TX、1000BASE-T
六类	UTP、STP	250	以太网 1000BASE-T
七类	STP	600	以太网 10G BASE-T

同轴电缆(coaxial cable) 一种外部由绝缘护套保护内部由两根同轴心、相互绝缘的圆柱形金属导体构成的传输介质。同轴电缆由四层材质构成:最内层的圆柱形金属导体由铜构成,铜导体周围是一层绝缘体,绝缘体外侧是由金属编织网和金属箔构成的网状导电体,最外层是绝缘护套,如图 3 所示。



图 3 同轴电缆

同轴电缆从用途上分可分为网络同轴电缆和视频同轴电缆,网络同轴电缆阻抗为 50 Ω,视频同轴电缆阻抗为 75 Ω,网络电缆又分细同轴电缆和粗同轴电缆。同轴电缆虽然存在受干扰较小、造价较低、传输距离较长等优点,但当一根电缆上一点发生故障时,故障影响到整根电缆上的所有机器,故障的诊断和修复都很麻烦,因此,同轴电缆已逐步为非屏蔽双绞线或光缆所取代。

常见的同轴电缆线有如下几种:

(1) RG-58/U 用于 10BASE2,阻抗为 50 Ω,直径为 5 mm,又称为“细同轴电缆”,需与 BNC 接头配合连接网络。

(2) RG-8/U 用于 10BASE5,阻抗为 50 Ω,直径为 10.3 mm,又称为“粗同轴电缆”,需与刺穿式搭接器配合连接网络。

(3) RG-59/U 阻抗为 75 Ω,直径 6.1 mm,常用做电视电缆。

(4) RG-62/U 阻抗为 93 Ω,直径 6.1 mm,早期的 ARCnet 使用这种电缆。 (崔建)

dianlu jiaohuan

电路交换(circuit switching) 通信网络中的一种交换技术。它在两个通信设备如电话或计算机之间经过几个交换节点创建一条直连的物理链路(电路),通信双方通过这条电路进行信息交换,如图 1 所示。

通过电路交换进行通信一般由电路建立、信息传输、电路断开三个阶段组成。例如在图 1 中 A 欲与 B 交换信息,则由 A 方向 B 方发起通信电路建立

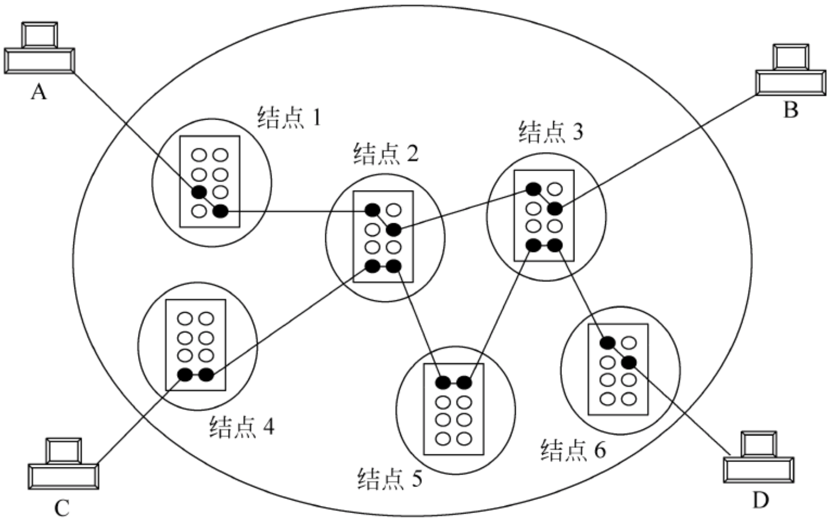


图 1　电路交换

请求，通过各交换节点的路径选择、各交换开关的接续，建立起一条由 A 到 B 的物理连接，A、B 双方独占这条连接并进行信息（数据或话音等）传输。传输完毕拆除连接，电路断开，A、B 双方的一次通信结束。

电路交换网中的核心设备是各级的交换节点或称**交换机**。常用的电路交换机有**空分交换机** SDS 和**时分交换机** TDS 以及空分、时分相结合的交换机。空分交换机的基本原理是采用纵横开关矩阵及多级纵横开关矩阵来形成输入输出交换路径，使得电路中的每一条路径在空间上与其他路径分开而互不干扰。在时分交换机中，交换则是使用时分复用（TDM）和时间槽交换（TST）来实现的。

参考文献

1. Stallings W, Van Slyke R. Business data communications. 3rd ed. Englewood Cliffs, NJ: Prentice Hall, 1998. 影印版. 北京：清华大学出版社，1998

2. Behrout Forouzen, et al. 数据通信与网络. 潘乞，朱丹宇，周正康，译. 北京：机械工业出版社，2000

（史美林　徐明伟）

dianqi yu dianzi gongchengshi xiehui

电气与电子工程师协会（Institute of Electrical and Electronics Engineers，IEEE）　源于 1884 年成立的美国电气工程师协会（AIEE）以及 1912 年成立的无线电工程师协会（IRE），1963 年 AIEE 和 IRE 合并成为 IEEE。经过几十年的发展，IEEE 已经成为世界上致力于先进技术创新的最大专业协会。IEEE 会员由工程师、科学家和相关专业人员组成，现在主要包括计算机科学家、软件开发人员、信息技术专业人士、物理学家、医学博士以及电气与电子工程领域之外的其他专业人员。IEEE 总部设在美国，地理上分成了 10 个管理区域，专业上分成 38 个学会和 7 个技术理事会。

IEEE 的较为活跃的标准化领域包括：设计自动化标准，局域网、城域网和个域网（个人区域网络）标准，信息保障标准，软件设计和开发标准，软件质量和管理标准，工业和商业设施标准，国家电力安全标准等。与计算机网络相关的标准主要是著名的 IEEE 802 系列标准，其中包括有关局域网的 IEEE 802.2 和 IEEE 802.3 系列标准，有关无线局域网的 IEEE 802.11 系列标准，有关局域网/城域网网络管理和网络互联的 IEEE 802.1、IEEE 802.10 和 IEEE 802.12 系列标准，有关无线个域网的 IEEE 802.15 系列标准，有关城域网的 IEEE 802.16 和 IEEE 802.17 系列标准等，这些标准已经成为事实上的国际标准。IEEE 网络标准主要侧重于物理层和数据链路层的技术规范，与 IETF 在网络标准方面的分工十分明确。

IEEE 标准化协会（IEEE-SA）负责 IEEE 的标准化工作。IEEE-SA 由会员选举出的董事会（BOG）管理，董事会监督下属分管 IEEE-SA 关键操作环节的委员会的工作，这些委员会包括 IEEE-SA 标准管理委员会（SASB），奖励和赏识委员会，公司咨询委员会，提名和任命委员会以及注册管理委员会，SASB 负责监督 IEEE 标准研制的过程，并且直接向董事会汇报。

SASB下设机构包括新标准委员会(NESCOM),负责确保设立的标准项目符合IEEE的范围和目的,具体的标准化工作由合适的IEEE学会或其他组织承担;评审委员会(REVCOM),对提交给SASB批准或接纳的标准提出批准或不批准的建议;程序委员会(PROCOM),对IEEE标准化程序的完善和改变提出建议;专利委员会(PATCOM),监督在IEEE标准中对任何专利和专利信息的使用;审计委员会(AUDCOM),负责例行检查,确保每个标准研制符合程序。

SASB还下设多个标准协调委员会(SCC),负责协调涉及IEEE多个技术委员会的标准制定工作,目前设置的SCC涉及电气绝缘、计量单位、消防、电子系统测试和诊断、能源存储、电源质量、自动读表、电磁安全和地球观察等方面。

IEEE的标准制定通常由负责与标准相关领域或相关内容的某个IEEE下属学会发起,通过项目授权请求书(PAR)正式提交立项申请。PAR被批准之后,就可以成立标准的工作组,创建和编写标准,工作期限通常不超过4年。

当工作组完成标准草案之后,可以进入投票环节。投票在标准制定的工作组之外进行,由标准发起方组织的投票组负责,投票组成员必须是IEEE-SA的会员,投票组包括制造商、用户、政府和其他利益团体成员(例如咨询顾问等),每个利益团体人数都不会超过半数。如果回收到75%的投票,并且回收票中的75%都是“赞成”票,则标准通过;如果收回的30%都是“弃权”票,则投票失败。

投票通过后,工作组向REVCOM提交标准草案及其支撑材料。REVCOM在审阅后,向标准管理委员会提交批准或不批准的建议,最终由标准管理委员会决定是否批准该标准。被批准的标准将公开发行,有效期10年。批准的标准在10年之后必须修订或撤销。

参考文献

1. http://www.ieee.org/

2. http://standards.ieee.org/　　　　(沈苏彬)

dianyuan ceshi

电源测试(power supply testing)　对电源的电性能、安全性、电磁兼容性、可靠性等进行测量,以确认被测电源的性能及其质量品质。

电源技术参数

(1)输入端　输入电压(交流或直流)、输入电流(有效值)、输入电流正负峰值、输入功率、功率因数、效率。

(2)输出端　直流输出电压、直流输出电流、输出功率、输出纹波(峰峰值)、交流输出电压(有效值)、交流输出电流(有效值)、波形失真率。

测试项目

(1)输入输出测试　测量电源在稳态状态下的输入及输出特性。

(2)动态测试　测量电源在动态负载情况下的输出特性。

(3)负载调整率测试　测量电源的一组特定的输出在三种不同负载情况下的响应,此时其他各组输出负载保持不变。

(4)交互效应测试　测量电源的一组输出在其他各组输出负载变动情况下所产生的效应。

(5)电压调整率测试　测量电源在三种输入电压情况下各组输出响应。

(6)组合调整率测试　在三种不同输入电压及负载组合条件下,测量电源的输出特性。

(7)开机及关机时序测试　测量电源在开机及关机瞬间各组输出的暂态响应。

(8)输入叠加噪声干扰测试　检测电源抗干扰能力。

(9)短路保护测试　当电源的一组输出对地短路后,测量其他各组的输出响应。

(10)过载保护测试　在过载条件下,测量电源各组输出响应。

(11)过压及电压过低保护测试　检测电源在输出电压过高或过低情况下的保护功能。

(12)波形失真率测试　对直流变换成交流的器件或设备,测量电源输出中高次谐波所占的百分数。

(13)耐压绝缘测试　测量电源输入对地和输出对地的绝缘性能。

(14)机壳接地电阻测试　测量电源外壳的接地电阻是否满足相关规范的要求,以避免漏电及触电危险。

(15)电磁兼容性测试　按照电磁兼容性的有关标准,检测电源满足电磁兼容性要求的能力。按照不同的使用要求,测试项目可包括雷击、浪涌、静电放电、电快速瞬变脉冲群、电流谐波、电压跌落、电压瞬变及短时中断、电压起伏和闪烁、辐射电磁场、传导干扰及辐射干扰等诸多方面的要求。

(16)老化寿命测试　高温及长时间满载条件下对电源进行老化测试。

（17）环境温度适应性测试　测量电源在规定的最高和最低工作温度情况下各组输出响应。

测试方法

早期电源测试是采用人工测试方法。测试者根据被测电源的类型、输出种类、技术要求确定其测试项目，并依此利用必要的仪器仪表、设备器材构成测试回路，进行人工观测，测试结果容易受各种因素影响，客观性和一致性较差。后来，一些电源专业厂应用电源测试系统进行自动测试。测试者根据所确定的测试项目编制测试程序，测试时测试系统在程序控制下实施对电源进行自动测试，显示并打印测试结果。测试快速、准确、高效、全面。

（张崇武　韩明）

dianyuan jicheng dianlu

电源集成电路（integrated circuits in power supply）　为计算机及其部件提供直流稳压供电的集成电路。直流电源中，整流、脉宽调制、稳压等功能的实现都用到了电源集成电路。

早期计算机直流稳压电源都是用分立元件制成的，体积较大，电源本身的耗电也多。后来，出现了小规模集成的电源专用组件，如整流桥堆、三端稳压器、脉宽调制控制器等。随着电子设备的微型化，电源集成电路的功能不断丰富，除常见的稳压器外，出现了多种不同功能的电源管理器件，如支持多种电压输出的器件、实现直流升压或降压的器件、专用于电池充电的器件、提供掉电保护的欠压/过压检测器件等。

整流桥堆　将整流二极管按要求封装在一个壳体内。有半波整流、全波整流等品种。

线性集成稳压器　当输入电压变化时，通过调节调整管的压降或电流，保持输出电压不变的模拟电路。它一般采用稳压负载和调整管串联的稳压方式。三端稳压器是线性集成稳压器的典型代表，有固定式和可调式两种，具有体积小、使用方便的优点。三端稳压器的缺点是在电源功率较大时，效率不高。为了改善电源效率，出现了低压差线性稳压器，用于输入/输出电压差较小、输入电压变化不大的场合。另外，还有多端可调稳压器、大电流低压差线性稳压器等形式。

开关集成稳压器　通过调节功率管的导通和截止时间来保持输出电压不变的稳压电路。它有脉冲宽度调制（PWM）、脉冲频率调制（PFM）等形式，PWM 用得最多。PWM 的工作原理是：当直流稳压电源的输出电压值偏高时，从 PWM 控制器送出的脉冲宽度变窄，从而将直流电源的输出电压幅度适当调低；反之，当输出直流电压偏低时，PWM 控制器送出的脉冲宽度就变宽，使直流电源的输出电压适当回升。PWM 控制器还可对电源及负载提供过流保护和过压保护电路。开关集成稳压器的优点是电源效率高、输入电压范围大、容易产生多种类型的输出电压等。

DC/DC 变换器　把输入的直流电压转换成另一种直流电压后输出的器件。DC/DC 变换器按输入/输出特性，可分为升压型、降压型、电压反转型和多功能型，按工作原理可分为电感式和电荷泵两类。DC/DC 变换器主要用于在板级部件上产生不同于输入电源的新电压，或者在用电池供电的便携式设备上产生必要的多种电源电压。

AC/DC 变换器　直接将输入的交流电压变换成直流电压后输出的器件。其核心通常是集成化的 PWM 控制的开关电源，具有体积小、重量轻、使用方便的特点。

DC/AC 变换器　把输入的直流电压变换成交流电压（通常是市电）后输出的器件，也称为逆变器，主要用在基于蓄电池的备份电源、不间断电源中。

充电控制集成电路　为多次重复使用的可充电电池充电的专用电路。充电控制集成电路中，除了直流稳压电源外，还包括恒流/恒压控制电路、电池识别电路、电池电压检测电路、电池温度检测电路、放电控制电路、充电电流检测电路、充电状态指示电路等部分。广泛用于各类便携式电子产品中。

电源监控电路　用于监控稳压电源工作状态，在失常时提供应急保护的集成电路，如过压/欠压检测电路、掉电保护电路等。

电压基准器件　为电路的其他部分提供基准参考电压的器件。早期采用稳压二极管或集成稳压器作为基准电压，现在大都采用特殊半导体工艺制造，具有温度系数小、精度高等特点。

参考文献

黄继昌，等. 电源专用集成电路及其应用. 北京：人民邮电出版社，2006

（林兼　唐志敏）

dianzhishu

电纸书（e-paper based book）　一种类似纸质出版物的阅读器。电纸书集多媒体信息内容、存储介质和显示终端于一体，将数字化的文字、图片、声音、影像等内容信息，通过基于电子墨水的电子纸显

示技术和声音合成等方法,将内容展示在一个类似纸质书大小的手持式阅读器上。人们可以像阅读传统纸质书一样地阅读电纸书,具有利于环保、辐射小、耗电低、不伤眼睛、存储容量大、便携等特性,是代替传统的纸质出版物,实现办公无纸化的新选择。

电纸书是电子书的一种展示形式,也是电子书的发展方向。电子书是指将文字、图片、声音、影像等信息内容数字化的出版物以及内置或下载数字化文字、图片、声音、影像等信息内容的集存储介质和显示终端于一体的手持阅读器。电子书不仅在电纸书上展示,还可以在 **PC**、平板计算机、手机、**个人数字助理**(PDA)等设备上展示,也就是说可以用计算机或手机等终端看电子书。电子书阅读器是专门用来浏览电子书的设备。电子书阅读器早在 20 世纪 90 年代就已经出现,个别国内公司在 2001 年已经推出了电子书阅读器产品,但是销售范围一直不大,主要集中在企业客户。2007 年 11 月,全球最大网络书城亚马逊(Amazon)推出电子书阅读器 Kindle,不到六个小时,6 in 屏幕的 Kindle 就销售一空。2009 年,第二代产品 Kindle2 问世两个月就销售了近 30 万部。亚马逊电子书的成功,引发了全球性电子书市场的迅速发展。为了推动电子书的进一步发展,成立于 1997 年的麻省理工学院下属的美国 e-ink 公司专注于电子纸和电子墨水的研发,该公司的电子超薄显示器技术代表了当前业内的最高水准。e-ink 公司所研发的电子纸张,表面看起来与普通纸张十分相似,可以像报纸一样被折叠卷起,奠定了电纸书的基础。

主要技术特点

(1) 电纸书一般使用由两片基板组成的电子纸张作为显示屏,上面涂有一种由无数微小透明颗粒组成的电子墨水,颗粒由密封在液态微胶囊内的带正、负电的许多黑色与白色粒子组成,不同颜色的带电粒子随施加电场的不同,而朝不同的方向运动,在显示屏表面呈现出黑色或白色,达到瞬间改变显示图文的目的。白色颗粒到达预定位置后,即使撤掉电压,仍会保持在原位置,无须持续加电,只有在变换内容即画素颜色变化时(例如从黑转到白)才耗电,关闭电源后显示屏上的画面仍可保留,异常省电。功耗是同尺寸大小薄膜晶体管(TFT)液晶显示屏的千分之一。

(2) 电子书阅读器按屏幕材质分为液晶屏电子书和电子纸屏幕的电纸书两大类。电纸书与液晶屏幕的电子书相比,具有以下特点:①更轻薄、无辐射、可长时间阅读、无闪烁、字号缩放自如、不伤眼睛,在阳光照射下不反光,可在阳光下阅读,阅读视角可接近 180°。②具有资料“记忆性”,只有画面异动时(例如由黑转到白)才耗电,关电后资料仍可留存在屏幕上,更为省电。③反射率高于反射式液晶屏,更接近报纸水平,对比度优于一般报纸,显示效果逼真。有些厂商的产品可达到影印用纸水平。④早期大部分以黑白方式显示文字内容,全彩或动画功能较差,应答速度普遍较液晶屏慢。⑤部分电子纸阅读器具有可挠性,或可卷曲性,而且字体大小可随意缩放。⑥目前成本较高。

(3) 长期以来,纸张是信息交换的主要媒介,但图文内容一旦印在纸张上,就不能改变,成为油墨、纸张复制工艺的最大缺点,难以满足现代社会信息快速更新对复制工艺的要求。电子纸具有的高分辨率和通过电场瞬间改变显示内容的特性,有效克服了上述缺点。而且显示材料可做成很轻、很薄,可弯曲,表面结构、阅读感受与纸张类似,加上电子墨水的功耗极低,电子纸可重复使用,使之成为新一代纸张的替代品。

(4) 支持格式多,电子书阅读器从最初支持单纯的 TXT 格式以及厂商的格式,到现在电子(纸)书阅读器支持大多数的图书格式,比如 TXT、JPG、BMP、HTML、PDF、DOC、WPS、CEB(Chinese eBook)、NLC(中国国家图书馆的电子图书格式)、OEB(Open eBook)等,有的电子(纸)书阅读器还可以支持 RAR、ZIP、PPT 等格式。

(5) 节能环保、一书多用。目前电纸书通常采用可充电锂电池供电,一次充电,开机状态可连续待机 15 天以上,在智能电源管理技术支持下,可连续翻页 7000 次以上。一本电纸书可以装载上千本电子版的图书。大多数产品可以扩充 SD 卡或 CF 卡等大容量存储器。一个扩充了 8 GB 存储卡的电纸书可以存储 40 亿个汉字,足以充当一个小型的移动图书馆,可节省大量纸张。一般电纸书阅读器的重量仅 100 至 200 克,机身轻巧便携。此外,通过网络下载或搜索互动,能获取大量的阅读资料,实现作者与读者的网络互动。读者尚可根据需要订制电子书,使个人出版成为可能。

发展趋势

电纸书阅读器目前处于初级发展阶段,电纸屏主要采用黑白、静态显示方式,存在只支持黑白显示、难以折叠和刷新速度慢等问题。今后电纸书的发展方向是开发全彩色动态显示、双面显示、多屏重

叠阅读显示技术，改进屏幕刷新速度，并提升产品的可用性和无线服务能力，形成以实用性强的彩色柔性技术为特征的新型电纸书阅读器，促使纸质书籍逐渐退出历史舞台，开辟通过网络和电纸书进行传播的发行渠道，这将引发出版发行业的结构性变革。电纸教科书的出现将会掀起教育行业的一大变革，不超过200 g重和不超过1 cm厚的电纸书将包含所有课本内容，孩子们不再需要背负沉重的书包上学。未来的电纸书还将引入**平板计算机**和**上网本**的部分功能，使电纸书具有更强的联网和网络搜索的能力。同时电纸书还将包含电子本的部分功能，即电纸书阅读器不但能“看”，还能“写”，具有原笔迹手写功能，可以对图文进行批注，还可以实现手写记事、手写邮件、远程签名等功能。

总之，电纸书阅读器是一种具有很好发展前景的产品，对出版发行行业的结构调整将产生重大影响。但是基于电子纸的显示技术仍需作深度开发，必须进一步提高电纸书阅读器的质量和销量，以大幅度降低成本，使电纸书像手机一样成为人们的必备工具。

参考文献

1. http://www.einkhome.com

2. http://en.wikipedia.org/wiki/E-book_reader

3. http://www.dianzhishu.com/news/20101011-1106.html

（韩承德）

dianzi shangwu xitong

电子商务系统（electronic commerce system） 综合运用信息技术，支持企业通过**互联网**、外联网、内联网进行商务运作及其有关信息交换与管理的**应用软件系统**。电子商务系统的作用在于改善企业与贸易伙伴间的协调和协作方式，改变传统商务活动中生产者、零售商和消费者之间分隔的层次关系，从而对于提升商业经营效率，提高服务质量，增强企业对市场变化的快速反应能力，乃至影响人们的消费方式和生活方式，都有重大的经济意义和社会效益。

电子商务的概念可以追溯到20世纪70年代的电子资金转账，银行通过加密网络简单地交换账户信息。70年代后期和80年代，电子数据交换技术（EDI）在一些发达国家逐渐形成一定规模，并很快引起了全球范围进行所谓“无纸贸易”的热潮。90年代开始，互联网在全球迅速发展，许多企业基于其内部资源的优化整合及其客户关系管理，通过**计算机网络**和商务软件，搭建企业及其产品的上下游供应链，开展了网上发布、网上查询、网上交易和网上支付等商务活动。1997年11月，在法国巴黎由国际商会召开的世界电子商务会议上对电子商务的概念进行了探讨。1999年12月在美国旧金山公布了世界上第一个互联网商务标准。21世纪以来，随着信息网络技术和现代服务业的发展，包括中国在内，全球的电子商务系统有了极大地增长，并形成了规模经济，成为当今信息网络服务和网络经济的一种重要商业模式。

电子商务系统应用范围广泛，因此有很多不同的分类方法，其中最基本的是从交易对象来分类：①企业与企业之间的电子商务（B2B）。供求企业之间以及协作企业之间利用网络交换信息、传递各种票据、支付货款，实现商务活动全过程的电子化。②企业与消费者之间的电子商务（B2C）。其典型应用是网上购物，实现电子化销售。③消费者与消费者之间的电子商务（C2C）。指由电子商务网站提供商品类目、拍卖或论坛，允许个人进行买卖业务，实现消费者之间的自由交易。还有支持政府网上采购的企业与政府之间的电子商务（B2G）等。

电子商务系统涵盖信息流、资金流和物流三个方面的业务，主要有：与销售业务直接相关的信息交换、售前和售后的信息服务，与银行业务有关的基于网络的电子支付，与运输业务有关的发送、跟踪等信息管理或数字产品的传送管理。

电子商务系统涉及的信息技术主要有：①系统架构　为有效支持电子商务业务的全过程，电子商务系统通常采用面向服务的体系架构SOA，相应的网络体系结构一般采用分层的多级服务器的浏览器-服务器结构，即客户端（浏览器）-内容发送网络-Web服务器-应用服务器-数据库服务器。②数据存储　数据是电子商务系统的核心，一个大规模电子商务系统往往需要存储海量的多结构数据，包括面向在线服务的集中式或分布式数据库系统和分布式对象存储系统等的数据存储与整合以及面向离线数据处理的数据仓库系统的构建。③信息管理和展示　对海量的商品信息进行有效的组织和管理，使用户更容易访问到有价值的商品，并通过多媒体交互等展示技术让用户更好地观察甚至体验到虚拟而真实的商品。④商业智能　运用智能搜索、联机数据分析和数据挖掘等技术，尽可能理解用户的请求，尽快找到用户想要的商品，并通过整合各类数据和统计报表，寻找和发现各类商务信息内在的相关性，

以进行商品的个性化推荐、对象排序、热点分析以及支持高层的商务决策等。⑤系统可靠性、可用性和扩展性　运用功能分离、关键数据备份、负载均衡、多级缓存和横向扩展性等相关技术以及身份识别、访问控制、数据安全、通信安全、加密、证书和签名等安全技术,既支持电子商务业务的快速和动态成长,又确保系统高可靠高可用高效率运行,既保护整个信息交换、电子支付和交易过程的安全,也保护个人隐私。此外,还有运用即时通信等系统,让消费者和商家进行直接交流和在线的客户服务的消息通信技术以及通过传感网来采集数据,通过射频识别(RFID)技术来跟踪物品,支持货物在物理空间中的识别、管理和运输的物联网技术等。

随着服务化技术、虚拟化技术、移动通信与新一代互联网技术、图形图像与视频处理技术、**虚拟现实**技术和智能软件技术等的发展和不断成熟,电子商务系统的内涵正在不断发展。作为一种面向广大消费者、众多商家和生产者的大规模分布式商品信息资源的查询、分析与服务系统,**云计算**及其服务模式将成为电子商务系统的发展趋势。

参考文献

http://en.wikibooks.org/wiki/E-Commerce_and_E-Business

(王坚　吴泉源)

dianzi sheji zidonghua

电子设计自动化(electronic design automation, EDA)　利用计算机来辅助设计电子器件、电路、装置和系统的技术。现代计算机或微电子器件从总体方案的论证(系统级)到指令序列的论证(行为功能级)、逻辑设计、功能验证、布局布线和测试等全部采用**计算机辅助设计**(CAD)、**计算机辅助制造**(CAM)、计算机辅助测试(CAT)技术,统称为电子设计自动化(EDA)。应用EDA技术,可以显著减少从设计到产品的成本和时间,而且还能解决手工无法解决的电子产品设计中日益增大的复杂度问题。因此,EDA的应用和发展,是现代计算机和超大规模集成电路(VLSI)发展的重要基石。

电子设计自动化始于20世纪50年代中期。当时计算机使用的还是晶体管,为了满足军用计算机的合同期限,各主要制造商都开始用计算机辅助设计。1956年提出了第一个辅助系统,用于逻辑方程检查、电路扇入扇出核对、机架布线长度计算、接线表编排以及底板布线等工程设计。这类始于计算机公司内部的EDA研发在60年代获得了迅速的发展,并很快受到大学研究人员的重视。进入70年代,EDA在计算机工程设计方面(如逻辑划分、布局和布线)获得了成功的应用,一些用于印制(电路)板(PCB)和集成电路设计的EDA系统开始形成商品。EDA的技术研究领域也从替代工程设计中烦琐、重复、高强度的人工工作,扩展到高层次的设计自动化问题。80年代以后,EDA技术逐步发展成熟,其应用覆盖了电子系统设计的各个层次,从概念设计、行为综合和模拟、逻辑设计、工程设计,直到热分析设计等。EDA也成为一个独立的产业,其产品(主要是各种EDA软件和工具)运行在微机、工作站、服务器等多种平台上,应用于集成电路设计、计算机部件设计、电信设备设计及其他复杂电子装置设计中。

现代EDA技术涉及从电子系统到基础器件、从逻辑功能到物理工艺的各个层面。下面以典型集成电路设计流程中的主要阶段为例,简述EDA技术的主要内容。

(1) **系统建模**　基于电子系统级(electronic system level, ESL)语言(如systemC, systemVerilog, C/C++等),开发待设计电路的系统级行为模型。通过模拟,不仅可以检验该模型是否满足电路设计规范的要求,而且可以检验待设计电路与其周边硬软件环境的相容性和互操作性,达到芯片与周边硬件(包括PCB)并发设计、硬件与软件协同设计的目的。EDA系统对电子系统级建模的支持是近几年才开始成熟的。在早期甚至当前的许多设计中,ESL模型都由人工转换为基于**硬件描述语言**(如Verilog, VHDL等)的寄存器传输级(register transfer level, RTL)描述。

(2) **设计验证**　检验设计是否符合规范要求,是产品开发过程中的重要阶段。因为在整个EDA流程中,设计会呈现系统级、寄存器传输级、门级、晶体管级等多种不同形式,所以设计验证贯穿于整个流程,通常占用了三分之二以上的设计资源(人员、设备、时间等)。设计验证的手段主要有逻辑模拟(包括硬件仿真)和电路模拟、功能验证、形式验证等。

(3) **逻辑综合**　把电路设计从行为级描述转换为结构级描述。通常意义上的逻辑综合指的是在给定的延时、面积等约束条件下,把基于VHDL或Verilog语言的电路RTL描述转换为逻辑门的网络(门级网表),其中涉及很多优化工作,目的是减少器件和连线冗余,进而减少芯片面积、缩短逻辑链延迟、

降低成本。把系统行为或算法描述转换为功能模块间连接关系或 RTL 描述的过程,有时称为高层次综合。

(4) **可测性设计**　为了测试复杂电路的正确性,不仅需要控制其输入端、观察其输出端,而且需要控制和观察电路中间的某些节点。可测性设计就是通过引入少量额外电路,增强电路中间节点的可控制性和可观察性,从而方便电路测试、扩大测试覆盖率、减少测试成本。常用的**可测性设计**技术有内部扫描设计和内建自测试两种,前者主要用于时序电路中的寄存器,后者主要用于片内存储器。

(5) **物理设计**　将表示为逻辑门网络或晶体管网络的电路转换为由多边形组成的几何图形(即用于制造掩膜的电路版图)的一系列过程。**物理设计**的主要步骤包括版图规划、布局布线、时钟树规划、物理验证等。由于半导体工艺技术的飞速发展,现代集成电路的特征尺寸持续缩小、供电电压不断降低,而管芯尺寸、封装密度、工作频率却在不断增大,物理设计需要考虑诸多因素:大量数字模拟或混合信号模块的集成、系统互连/总线设计、性能面积功耗的优化、信号完整性等,而且物理设计工作必须能够在有限的时间内处理大规模的设计,以满足产品上市时间的要求。

(6) **测试诊断**　测试指检测电路故障的存在;诊断则是确定故障在电路中的位置。一方面,需要根据可测性设计的安排,生成覆盖率高的测试向量,进行电路的测试和验证。另一方面,需要与自动测试仪(ATE)配合,基于测试向量开发高效率的测试程序,支撑芯片的量产测试。测试压缩是这里的关键,它不仅有助于缩短测试时间、减少测试成本,而且可以减少内建自测试需要的测试码存储量,缩小芯片面积。

(7) **低功耗设计**　各类有助于降低集成电路功耗的设计方法的统称。集成电路的功耗主要分为动态功耗和静态功耗两大部分,前者是电路工作时产生的,包括跳变功耗和短路功耗,主要与晶体管的开关频率和供电电压有关;后者主要与漏电流有关,只要通电就存在。可以从系统、算法、结构、逻辑、电路等多个层次着手降低电路的功耗。在芯片设计层面,降低动态功耗的主要方法是门控时钟,降低静态功耗的主要方法是门控电源。

(8) **热分析建模及模拟技术**　指建立电子元件通电发热数学模型,应用模拟软件分析,显示集成电路芯片或电子设备装置的热分布状态,从而帮助设计师找出热点,以便改进布局设计和封装方案。

国内对 EDA 技术的研究始于 20 世纪 70 年代中期,主要在两大领域展开研究。一是印制线路板计算机辅助设计和制造(PCB-CAD/CAM);二是集成电路 CAD 系统。经过数十年的研究、开发和攻关,我国已拥有自己的集成电路 CAD 系统,并在工厂投入实际使用;拥有自己版权的多种 PCB-CAD/CAM 系统,在电子行业获得应用。这对普及 EDA 技术也起到了促进作用。目前,我国正继续加强这方面的研发投入,推动相关系统向商品化、集成化方向发展。

电子设计自动化的最终目的是使电子系统设计、制造、测试实现自动化。随着集成度的不断提高和片上系统的不断发展,芯片规模和复杂度不断上升,给 EDA 技术带来了新的问题和新的挑战,如设计和验证的生产率、模拟和混合信号电路的协同设计、IP 核的可重用和测试、功耗管理和低功耗设计、可制造性和制造测试、硅后验证和产品可靠性,等等。这些挑战也正是 EDA 技术进一步发展的动力。

参考文献

1. 刘明业. 数字系统设计自动化. 北京:电子工业出版社,1994

2. Wang L T, Chang Y W, Cheng K T. Electronic design automation: Synthesis, verification, and test. Morgan Kaufmann Publishers. 2009

(潘雪增　唐志敏)

dianzi youjian

电子邮件(electronic mail, Email)　在发送者和指定的接收者之间,利用计算机网络进行的文本、数据、图像或语音(言语)等信息的非交互式通信。在计算机网络众多的服务中,电子邮件是使用最广泛的服务之一。电子邮件系统既是一种通用的网络应用,也是一种为其他应用所使用的基础设施。

在电子邮件发送前,每个用户必须有一个电子邮箱来存放邮件。每个邮箱有一个唯一的邮件地址,它分为两部分:第一部分标识用户的邮箱,第二部分标识邮箱所在的计算机。一种广泛使用的格式是 mailbox@computer,这里 mailbox 是一个指明用户邮箱的字符串,而 computer 是一个指明邮箱所在的计算机的字符串。

简单邮件传送协议(SMTP)是 **TCP/IP 协议**集中用于主机之间相互传送邮件的标准协议。SMTP 采用了 RFC 822 定义的电子邮件发送的文本报文格

式,包括信封和报文内容。信封包含的是各种用于完成邮件传输所必要的信息,传统的 SMTP 仅局限于传送简单的文本报文。

通用 Internet 邮件扩展协议(MIME)是 RFC 822 框架结构的扩展,可传输多媒体信息、包含各国语言字符的文本数据、可执行文件或其他二进制文件等。

(胡道元)

dianzi zhengwu xitong

电子政务系统(electronic government system) 支持政府管理社会公共事务的**应用软件系统**。电子政务系统区别于政府内部办公信息系统的主要特征是在公共事务存在与演化的整个生命周期中,支持超越时间空间及部门分隔、强调决策预见性和公众参与性。

电子政务概念的内涵,经历了一个发展变化的过程。20 世纪 60 年代,发达国家开始将现代信息技术应用于政府的业务,实现业务过程的电子化。计算机在政府部门的应用从一般的数据处理开始,80 年代前后,逐步向**办公信息系统**、**管理信息系统**和**决策支持系统**发展。随着**互联网**技术的发展及在政府公共管理中的应用,1992 年美国首次提出电子政务的概念,并很快得到世界各国政府的积极响应。中国于 90 年代后期开始电子政务建设,取得了一定成效。目前,电子政务系统已成为发达国家政府信息化最重要的组成部分。它对于创建一个智能化、高效率、公开透明和低成本运行的公共管理服务型政府正发挥着越来越大的作用。

电子政务系统的核心是政务软件,其主要内容有:①监测分析 诸如能源监测、资源监测、环境监测、物价监测、质量监测、经济运行监测、金融市场监测等,通过及时获取和分析以上经济社会系统状态运行和变化的动态信息数据,支持公共管理决策、评价公共管理绩效。②行政执法 诸如市场准入、税收征管、交通管理、行政处罚等,通过规范经济社会系统秩序,支持过程留痕,规范和制约政府部门依法行政和责任追究。③公共财政 诸如财政预算、政府投资、转移支付、科教文卫投入等,通过规范财政预算形成过程和开放公共资源配置参与,支持财政管理公开透明,努力保障社会公共福利的最大化。④内部管理 诸如财政审计、行政监察、效能评估等,通过对公共管理业务的痕迹真实性、过程合规性、运行效益性的监督评价,落实政府部门依法自律的内部控制机制。⑤政策决策 诸如财税金融、节能环保、产业发展、科技进步、公共卫生、社会保障等,通过汇聚决策信息资源、制定社会公共政策决策模型,拟定公共政策可选目标、落实多目标群决策机制,生成制度安排可选方案,支持民主、科学、有预见性的政策决策。⑥开放参与 诸如信息公开、意见征询、听证会议等,通过信息收集、聚类分析和信息发布,保障公众对公共事务决策过程的知情权和参与权。⑦应急处置 诸如抗洪、防疫、救灾、隐患发现、风险预警、应急指挥等,支持应急预案自动落实和过程留痕,最大限度地减轻紧急事件所造成的社会利益损失。

电子政务系统的基础是网络。网络基础设施通常必须满足内、外网络体系。内网除了运行政府办公信息系统外,需要为外网的公共管理服务进行必要的政府内部信息资源的整合、处理与优化。外网运行社会公共事务管理服务应用系统,为社会、企业乃至公民个人提供政府信息服务,实现网上发布、网上查询、网上申报、网上审批、网上决策和网上监察。对于某些重要行业,还可以设置专网,用于运行相关的行业业务管理服务系统。统一的信息安全支撑体系应该贯穿于从网络层、业务层乃至应用层的所有环节。

电子政务系统是以公共管理业务的需求为导向的。当前,呈现出如下主要特点和发展趋势:①分布式信息资源组织 公共事务信息是自主分布的,电子政务系统应该主动地保持和利用物理分布并且存在语义和结构差异的公共事务信息,通过松散耦合和虚拟化等技术实现信息资源的集成与共享,一般不采用物理上集中、逻辑上紧耦合的信息资源体系。②适应性应用开发方法 从社会公共事务系统即公共管理的对象系统出发,针对公共事务系统运行的关键因子,面向公共事务管理的高度适应性,设计公共管理业务模式,进行电子政务系统的顶层设计,进而利用分布计算中间件和业务流程管理等技术,进行管理业务应用系统的定制化开发和部署。③**云计算**服务模式 电子政务系统是一种面向公众的大规模分布式信息资源查询分析应用系统,随着虚拟化、服务化和资源聚合等技术的发展和不断成熟,**面向服务的体系架构**和云服务应用模式将成为电子政务系统的发展趋势。

参考文献

Wikipedia. E-Government. http://en.wikipedia.org/wiki/EGovernment

(张文燚 吴泉源)

dingxing fangzhen

定性仿真(qualitative simulation) 以模仿人类思维方式或某种定性理论推导系统的定性行为描述,以实现系统定性建模,并处理仿真的信息输入、建模、行为分析和结果输出等的一类仿真技术(参见**计算机仿真**)。

1983 年美国 XEROX 实验室的学者 Seely Brown 和 John de Kleer 发表了论文"A Qualitative Physics Based On Confluence",首次提出关于定性建模理论并揭开了定性仿真研究的序幕。1984 年国际人工智能杂志第一次出版了关于定性问题的专辑,从此,定性仿真技术得到了迅猛发展。

定性仿真发展过程中形成了多种派别,其中比较有影响的是:Seely Brown 和 John de Kleer 提出的基于"流"的概念的理论;K. D. Forbus 的定性过程理论(QPT),B. J. Kuipers 的基于 QSIM 的定性仿真理论(Qs)以及 B. C. Williams 的定性代数理论等。以下介绍几种典型的方法。

归纳推理法 归纳推理法源于通用系统理论中的 GSPS(general system problem solve)技术。输入尽可能多的行为,通过归纳学习的方式,构造系统的定性模型,进行仿真研究。归纳推理法可以模仿人类固有的概括总结和学习的能力,处理观测数据辨识系统中的依赖关系,建立并优化系统定性行为模型,预测系统行为。

但是,这种方法需要采集大量的数据并处理和维护;而且,往往由于现实条件的限制,不能保证归纳的完备性。

非因果关系推理方法 非因果类方法是系统建模时不需要明确指出系统内状态变迁过程的因果方向。属于这一类范畴的方法有多种形式。典型的有:基于流(confluence)的概念以组元(components)为中心的方法;基于定性微分方程(QDE)以约束(constraints)为中心的方法,以过程(process)为中心的方法以及 TCP 时间推理方法等。

基于因果关系的推理方法 基于因果关系的推理方法无一例外地依赖于有向图(oriented graph)定性传递函数方法。Raiman O 在 1986 年提出 FOG 系统,通过表达变量关联的约束网络传播初始化信息,通过 FOG 的方法可以减少系统中的次要作用,继续保留系统中的主要作用。Ousson K 和 Trave—Massuyes 在 1992 年提出了结合系统的因果关系和深层次知识,将模型表示为两层约束,第一层是由因果图支持的局部约束层,第二层是由物理方程组成的全局约束层,可以克服按照因果关系建模无法表示全局性的约束关系等局限。

基于图表的推理方法 所谓图表推理就是通过模拟人类的感知方式和形象思维的途径,而不是利用传统的符号形式化方法进行推理。近些年来,许多定性推理方面的科学家都投身于图表推理的工作中,以期能够找到一套不利用符号进行推理的方法。一些学者对图表进行了定性可视结构的研究。图表是人类认知的形象描述,它并不受数学的形式限制,图表推理在定性推理和仿真领域占有越来越重要的地位,在定性领域有着广泛的应用前景。

基于数据的结构化建模方法 现实中很多系统,一般只能获取它的实测数据,而缺乏其结构知识。意大利科学家 Lilina Ironi1 提出了一套将定性理论用于定量建模过程的方法,在仅仅拥有实验数据的情况下,结合领域知识,利用定性推理的系统辨识方法,建立系统的结构化定量模型。系统辨识过程是由应用任务和先验知识构成模型空间,从模型空间中提炼出一个初等模型子集,经过测试并观测初等模型子集结构,不断改进此模型子集以得到一个定量模型。再通过对定量模型的评估,使模型不断细化直到通过评估而得到一个动力学系统模型。该模型对于数据的分析依赖于其智能数据分析框架。在获得观测数据之后,先对其进行操作,得到预处理数据;另一方面,基于定性推理的方法利用领域知识库和观测数据对模型几何形状进行辨识,再结合基本物理特征知识获得观测的物理特征评估,作为定性响应与预处理数据,结合领域知识从而完成系统辨识。这种方法已经成功应用于药理学实践中。

基于定性空间的推理方法 在用传统定性推理理论进行空间图形的推理描述时,缺乏形象的表现力,因此,发展出了基于定性空间的推理方法,即在进行空间推理的时候,利用方向、距离、拓扑序、连接性、边界、区域、形状等对空间实体进行定性描述。

相对于通常的仿真而言,定性仿真在处理不完备知识和"深层"知识以及决策等方面有其独到的长处。现在,定性仿真技术与物理、化学、生物学、社会学和经济学等学科相互渗透、相互结合、相互推进,在电力、交通、工程机械制造、生化、商业流通等领域得到了许多应用。

参考文献

1. Eberhart R C, Shi Y. Particle swarm optimization: developments, applications and resources. Proc Congress on Evolutionary Computation. Piscataway:

IEEE,21301:81-86

2. 邵晨曦,白方周. 定性仿真技术及应用. 系统仿真学报, 2004, 16(2)

3. 李文伟. 定性仿真方法发展与应用研究. 系统仿真技术,2008, 4(2) (王威 肖田元)

dingxing shikong biaoshi

定性时空表示(qualitative spatio-temporal representation) 时态、空间或时空知识的定性表示方法。

定性时空表示研究始于20世纪80年代,最初针对时态问题,后扩展到空间和时空问题。20世纪90年代,逐渐成为人工智能的重要研究子域。进入21世纪,研究重点转向复杂时空知识的定性表示。目前,定性时空表示主要应用于常识推理、规划、自治机器人导航、**自然语言理解**、**计算机视觉**、**地理信息系统**和**时空数据挖掘**等领域。

定性时空表示的研究方法主要包括代数方法和逻辑方法。逻辑方法参见**空间逻辑**,下面重点介绍代数方法。定性时空表示侧重研究时空关系和时空场景的定性表示问题。时空关系表示是对2~3个对象间的时空关系进行定性描述,具体分为时态、空间和时空关系表示。时态关系表示主要针对序数时间、区间时间、环时间和分支时间等,代表性工作是Allen的时间区间代数及其衍生理论(如Ladkin代数);空间关系表示主要包括块代数、区域连接演算(region connection calculus,RCC)、主方位演算(cardinal direction calculus,CDC)等方法,它们分别针对不同的空间维度(一、二、三维)、论域(点、线和区域等)和关系类别(拓扑、方向、尺寸和距离等);时空关系表示有定性轨迹(qualitative trajectory calculus, QTC)等方法。时空场景定性表示是基于定性时空关系对n个对象构成的场景进行定性描述,主要采用约束方法,其核心问题是解决定性约束满足问题(qualitative constraints satisfaction problem, QCSP)。区间代数、RCC基本关系的QCSP可在$O(n^3)$时间内判定,其全关系QCSP是NP完全问题,已找到了相应的最大可处理子集。CDC基本关系的QCSP可在多项式时间内判定。其他模型的QCSP尚未解决。实际问题中的时空场景往往需要综合考虑多种时空关系,此类问题称为联合约束满足问题(joint satisfaction problem,JSP),目前已开展了一些组合两个模型的JSP研究(如RCC分别与定性尺寸、CDC组合),尚缺少组合两种以上模型的有效方法。不确定性是时空信息的重要特性之一,定性时空表示中不确定性处理方法主要有宽边界、蛋黄模型、模糊集和粗糙集等方法。

解决诸如复杂空间、复杂时空对象和复杂时空关系等现实复杂问题是当前定性时空表示面临的主要挑战。此外,定性时空表示分别与时空数据挖掘、图像处理、语义Web等的结合研究也很有发展前景。

参考文献

1. Cohn A G and Renz J. Qualitative spatial representation and reasoning. In: van Hermelen F, Lifschitz V, Porter B, eds. Handbook of knowledge representation. Elsevier, 2008

2. Renz J, Qualitative spatial reasoning with topological information, LNAI 2293, Berlin: Springer-Verlag, 2002

3. Escrig M T, Toledo F. Qualitative spatial reasoning: Theory and practice. Amsterdam: IOS Press, 1998 (王生生 刘大有)

dingxing tuili

定性推理(qualitative reasoning) 从物理系统的结构描述出发,导出行为描述和功能描述,预测物理系统的行为,并给出因果关系的解释的一种非定量推理方法。

量的定性描述是有明确结构的一种非定量的、非精确的表示方法,如物理系统变化趋势的定性描述常是有效的。定性推理是源于物理现象的一种常识推理,1977年Reiger发表了第一篇定性推理的论文,1984年《Artificial Intelligence》杂志出版了定性推理专辑,刊登了D. J. Kleer, K. D. Forbus和B. Kuipers等人关于定性推理的奠基性文章,标志着定性推理开始走向成熟。随后,定性推理得到人工智能界的普遍关注,得以发展。

传统的定量方法是精确描述物理量间的关系,但不具备推理能力,物理量之间的因果关系湮没在精确的数值中。而定性方法具备推理能力,能表达物理系统的因果关系,通过局部传播能在较高层次上给出系统的宏观描述。

结构描述和行为描述可理解为定量方程及其解的抽象,功能描述是对实际物理系统行为表现的一种理解。定性推理的论域是离散变化的符号集,最简单的定性论域是{+,0,-},相当于把实数轴离散化为{(-∞,0),[0,0],(0. eps,+∞)},开域内

的值表现了定性一致的行为性质,而边界值反映的则是转化点。

定性推理的基本方法如下:

Envision 方法　系统的结构用一个定性方程组来描述,系统的论域采用符号集{+,0,-,?}。“?”的引入是为了保证运算的代数封闭性,表示不确定。系统的所有变量的一组取值构成一个状态,由初状态出发而得到的所有状态和状态转移构成了可能的展望空间。这种随时间变化的定性状态序列构成了系统的定性行为。

QPT 方法　系统用个体视图和进程来描述,个体是实际物理系统中存在的物理实体。个体视图是对个体及个体组合的抽象和描述。进程描述实际物理系统的变化。从实际物理系统的结构描述获得系统的行为描述就是推导实际物理系统视图结构和进程结构的过程。

QSIM 方法　系统的论域是被物理量的界标离散化的实数轴,结构描述由系统物理量之间的约束组成。约束是物理量之间的定性关系,本质上是一组定性方程。推理过程是从系统的初状态产生所有可能的定性后继状态,根据约束关系从中选出各种相容的定性状态。重复这一过程,可得到实际物理系统的定性状态变化过程。

因果分析法　定性推理方法倾向于模拟人的思维方式,在对物理系统的行为求解的同时,要对行为做出解释,包括基于约束的因果分析、因果顺序分析和约束网络中的因果分析等。

定性定量相结合方法　是在定性的基础上引入定量信息,二者结合进行推理。

定性空间推理方法　涉及刚体运动的几何性质,典型的如机器人运动的规划、导航。

定性推理有广泛的应用前景,如在诊断、设计等方面。

参考文献

1. Weld D, de Kleer J. Readings in qualitative reasoning about physical systems. Los Altos, CA: Morgan Kaufmann, 1990

2. 石纯一,廖士中. 定性推理方法. 北京:清华大学出版社,2002

(廖士中　石纯一)

dongtai guihua fa

动态规划法(dynamic programming approach)　一种一般的用于求最佳解的**算法设计**方法。动态规划法的基本思想是:在逐步由局部解扩展成全局最佳解的过程中,在每一步,首先列出各种可能的局部解,然后按某些条件,从这些局部解中挑选出那些有可能产生全局最佳解的结果,留下来在下一步继续扩展而扬弃其余。只有满足最佳原理的问题才可用动态规划法来求解。所谓最佳原理是指:不论前面的状态和策略如何,后面的最优策略只取决于由最初策略所确定的当前状态。通俗地说就是当前的最优策略与历史无关,只与现状有关。

特别地,对最短道路问题,假设从起点 S 到过渡点 I 的道路不影响从 I 到终点 G 的道路的选取,那么从 S 经 I 到 G 的最短距离是从 S 到 I 的最短距离与从 I 到 G 的最短距离之和。结果,在寻找从 S 到 G 的最短道路时,以下动态规划原理成立:从 S 到任何过渡点 I 的所有道路,除了从 S 到 I 的最短道路外,其他都可忽略。动态规划法的一种通用实现方法是:建立一个队列来记录目前已形成的所有从 S 到某些过渡点的道路,并按至今累计的代价对整个队列进行排序使代价最小的道路排在前面。然后检查队列中的道路,如果存在两条或多条道路达到某个公共结点,那么除了以最小的累计代价达到该公共结点的那条道路外,删除所有其他达到该公共结点的道路。最后扩展排在队列最前面的道路并将扩展出的所有新道路加入队列中并转上面的队列排序操作,直到排在队列最前面的道路是一条从 S 到 G 的道路。从上可以看出:如果存在许多达到公共结点的道路,那么动态规划将是高效的。用动态规划法求解问题时,有时设计的算法需要多项式执行时间(如矩阵链乘法问题为 $O(n^3)$);有时设计的算法需要指数执行时间(如旅行商问题为 $O(n^2 2^n)$);有时设计的算法需要指数执行时间但经数学分析后可优化成只需要多项式执行时间(如双处理机流水作业车间调度问题可优化为 $O(n\log_2 n)$)。

使用动态规划法求某个问题的“最小代价解”或“最大获利解”时,一般首先需要定义某种局部解(定义局部解的方式可能不唯一,不同的方式一般导致不同的算法)及其“最小代价”或“最大获利”。然后根据动态规划法,建立全局解的“最小代价”或“最大获利”与局部解的“最小代价”或“最大获利”之间的关系。在这样的关系式中,若原问题是求“最小代价”则对各种可能的分解全局解的方式的总代价取最小值,否则若原问题是求“最大获利”则对各种可能的分解全局解的方式的总获利取最大值。因为全局与局部是相对的,所以这样得到的关系是递推关系。当局部解的规模足够小时便直接计

算其“最小代价”或“最大获利”。这时,按规模从小到大的方式来组织计算,最终可以算出全局解的“最小代价”或“最大获利”。为了跟踪构造全局最佳解的过程,还需要为每个局部解定义一个量来记录其最佳的分解成更局部解的方式,一般记录的就是上述递推关系中使总代价达到最小值或总获利达到最大值的那种分解方式,如果存在多种方式同时使总代价最小或总获利最大,那么原问题存在多个最佳解。在按规模从小到大的方式同步为所有局部解计算出这个量之后,便可按自顶向下的方式构造出全局最佳解。

作为一种一般的算法设计方法,动态规划法可用于求解矩阵链乘法问题、最长公共子序列问题、最优多边形三角剖分问题、切棒问题、最短路径问题、最佳折半查找树问题、装配线调度问题、资源分配问题、多机系统的可靠性设计问题、流水作业车间调度问题、带权的区间调度问题、分段的最小二乘问题、子集和问题、整数背包问题、RNA 二级结构问题、序列比对问题、图中的负圈问题和旅行商问题等优化问题。

参考文献

1. Aho A V, Hopcroft J E, Ullman J D. The design and analysis of computer algorithms. Boston, MA: Addison-Wesley, 1974

2. Cormen T H, Leiserson C E, Rivest R L, et al. Introduction to algorithms. 3rd ed. Cambridge, MA: The MIT Press, 2009

3. Kleinberg J, Tardos E. Algorithm design. Boston, MA: Addison-Wesley, 2005

（殷建平）

dongtai lianjie

动态链接(dynamic link)　在目标程序装入或执行时将尚未链接的例程进行链接,并将程序中所有未解析的符号进行绑定的机制。

动态链接的基本原理是:链接编辑程序在链接时,只记录程序需要的例程和该例程在相关库文件中的索引,例程的链接和全局变量等某些符号的解析与绑定等工作推迟到程序装入或者执行时进行。这些 可动态链接的例程来自于专门的动态共享库,在 Windows 系统中称为动态链接库,在 Unix 类系统上称为动态共享对象。动态共享库中的程序格式和可执行程序本质上是相同的,包含数据和代码,不同的只是与位置无关,并加上了表示本身不可执行的特定选项。

与动态链接相对的是静态链接,在程序链接时就将程序和例程库中的相关例程经链接与符号绑定组合成一个大的程序段或可执行程序,属于早绑定。为了解决静态链接存在空间浪费、软件模块更新困难等问题,在 1964 年就出现了动态链接。它属于晚绑定,能够支持程序运行时动态选择需要装入的程序模块,从而不仅避免了静态链接存在的问题,还促进了程序插件技术的发展和应用。

在实现上,动态链接由操作系统的动态链接器(或称链接装入器)来完成装入时或者运行时多个目标程序的链接装入和重定位,包括把所有相关的动态链接库(含相关的例程库和其他软件库)中的目标程序装入到操作系统分配的地址空间,将程序中所有未解析的符号进行绑定,进行重定位,并最终产生机器上可执行的程序。

参考文献

1. Levine J R. Linkers & loaders. The Morgan Kaufmann Series in Software Engineering and Programming. 1st,1999

2. 俞甲子,石凡,潘爱民. 程序员的自我修养——链接、装载与库. 北京:电子工业出版社,2009

（黄春）

dongtai suiji cunchuqi

动态随机存储器(dynamic random access memory, DRAM)　(参见半导体存储器)

dongtai suiji cunqu cunchuqi xinpian

动态随机存取存储器芯片(dynamic random access memory chip)　一种可以随机存取,但必须周期性地对其存储数据进行动态刷新,以防数据消失的存储器芯片。动态随机存取存储器(DRAM)芯片的存储单元将数字信息以电荷形式存于电容上,由于漏电,电容电荷会缓慢消失,导致读出数据错误。因此必须周期性地进行充电刷新,以保证所存数据不致丢失。而断电时,DRAM 芯片所存数据将会消失,属易失性存储器。

分类　通用 DRAM 芯片按接口可分为非同步 DRAM 和同步 DRAM(SDRAM)芯片。作为基础的非同步 DRAM 芯片有快速页面模式(FPM)和数据扩展输出模式(EDO)等工作方式,但目前已很少使用。SDRAM 芯片的内核仍是非同步 DRAM,但增加了同步时钟输入,将地址、控制命令、数据输入输出的接口改为由时钟同步。按数据输入输出的速率与

输入时钟频率的关系，SDRAM 芯片又分为 SDR（单倍数据速率）、DDR（双倍数据速率）SDRAM 芯片以及 Rambus DRAM（RDRAM）芯片等。根据输入时钟频率范围的不同 DDR SDRAM 芯片有 DDR（时钟频率 100/133/166/200 MHz）、DDR2（时钟频率 200/266/333/400 MHz）、DDR3（时钟频率 400/533/667/800 MHz）SDRAM 芯片。通用 DRAM 芯片的封装、引脚排列和定义、内部构成和时序控制有 JEDEC 标准，以保证通用性。个人计算机（PC）的 CPU 芯片的工作频率、处理能力的迅速提高和图像、网络的应用不断扩大，对作为 PC 主内存的 DRAM 芯片的容量和速度的要求也不断提高。满足 PC 发展的要求成为推动通用 DRAM 芯片不断发展的主要因素。而无线通信、手持设备、影音图像等消费电子设备对高性能低功耗小体积的要求促使专用 DRAM 芯片的发展。这些专用芯片是在通用芯片的基础上通过降低功耗，采用特殊封装或复合其他功能而形成的。

存储单元和存储单元阵列　DRAM 芯片的存储单元电路种类很多。由于单管单元所需器件少，对于相同的芯片面积，用单管单元的芯片可容纳更多的存储单元，因此单管单元的应用最为广泛。

图 1 为 DRAM 芯片的单管存储单元示意图。每个存储单元由一个 MOS 管和一个存储电容组成。MOS 管用于地址选择，其栅极接行（字）选择线，源极接列（位）数据读出放大和写入驱动线，漏极接存储电容。用电容的存储电荷表示存储的数据。若电容充电，则电容上电压高，表示 1；电容未充电，则电容上电压低，表示 0。由于读出差分放大的设计，差分放大一端列线上的存储单元 1/0 的区分与一端列线上的存储单元相反。

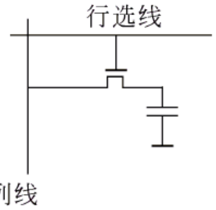


图 1　DRAM 芯片的单管电路

DRAM 芯片的核心是存储单元阵列。DRAM 芯片的存储容量为 $2^N \times M$ 位，即字数为 2^N，N 为确定每个字位置的地址的长度；M 表示输入输出数据的位数，即芯片数据的字长。这样就确定了所访问存储单元的数据位数。数据位数 M 通常为 4，8，16 或 32。为平衡阵列的行选择线和列线的负载，2^N 个存储单元排成 2^{N_R} 行 $\times (2^{N_L} \times M)$ 列的阵列。输入地址也分为行地址和列地址，N_R 和 N_L 分别为行地址和列地址的位数，$N_R > N_L$。随着 DRAM 芯片容量的增大，为减少行选线和列选线的电容负载，以保证读出时间，芯片内部结构分为 2^{N_B}（$N_B = 1/2/3/4$）个独立的存储体。存储体的地址（BA）为 N_B 位。初期 N_B 多为 1；随着容量的扩大，N_B 已增加到 3/4（即 8/16 个存储体）。每个体有各自的存储单元阵列和外围电路（行译码和驱动、列译码和读出放大/写入驱动）。地址总长度 $N = N_R + N_L + N_B$。芯片的存储容量为 $(2^{N_R+N_L} \times M) \times 2^{N_B}$。

为了减少地址输入引脚的数量，以降低封装成本和尺寸，DRAM 芯片均采用分时地址方式，即行地址和列地址共用地址输入引脚，按时序先输入行地址，后输入列地址。分时地址输入和相应的时序控制信号由专门的接口控制电路产生。

非同步 DRAM 芯片的组成　DRAM 芯片由存储单元阵列、行地址缓冲、列地址缓冲和译码驱动电路、读出放大和写入驱动电路、数据输入输出缓冲、再生地址计数与控制以及时序控制电路等组成。

非同步 DRAM 芯片无时钟信号输入，以行选输入 RAS#和列选输入 CAS#为基本时序控制。当 RAS#由高电平（未选）变为低（选中）时，下降边将行地址打入行地址缓存。经过译码后，将选中的一条行线驱动到高电平，其他行线仍为低电平。与选中行线相连的所有存储单元的存储电容上的电荷与相应的列线的寄生电容电荷再分配，使列线电平有所升高或降低，所存数据也受到破坏。由于存 0 与存 1 列线电平差别小，为了便于鉴别，读出放大电路采用差分放大。然后再鉴别所存储的数据是 1 或 0，并将其存入选中体各列的数据寄存器中；再按所存入的数据驱动列线，通过选中行的 MOS 管对电容充电（存入 1）或放电（存入 0）。这样将数据重新写入存储单元（恢复）。经过 RAS#-CAS#间的延迟时间 t_{RCD}，CAS#由高变为低电平，其下降边将列地址打入列地址缓冲。经列地址译码，将选中的列数据寄存器中的数据送到输出数据缓冲。由于选中行线的所有存储单元的数据均输出到列数据寄存器，因此 DRAM 可以选择快速工作方式，或称为页面工作方式。即在 RAS#变为低后，CAS#连续有负脉冲，访问已选中行的不同列的存储单元。页面的大小为 2^{N_L}。DRAM 芯片常用的数据输出方式有快速页面（FPM）（当 CAS#为低电平时，经列地址读出时间 t_{cac}，输出数据）与数据扩展输出（EDO）（CAS#的下降边经列地址读出时间 t_{cac}，锁存数据输出）这两种

快速工作方式。RAS#为低电平的最小宽度为 t_{RAS}。在下次输入行地址前 RAS#必须变为高电平,使所有行线处于未选状态,以保证列线电平恢复到行线选中读出前的稳定状态,这样新的选行命令才能读出正确的数据。RAS#为高电平的最小宽度为最小预充电时间 t_{RP}。最小 RAS#周期 $t_{RC}=t_{RAS}+t_{RP}$。行选读出时间 $t_{RAC}=t_{RCD}+t_{CAC}$。

同步 DRAM 芯片的组成 SDRAM 和 DDR SDRAM 芯片的结构基本相同,其内核均为非同步 DRAM。差别主要在存储体数及体的容量,数据的输入输出预取处理电路(DDR 预取 2 位、DDR2 预取 4 位、DDR3 预取 8 位)和数据输入输出的同步时钟及传输匹配电路。图 2 为 4×16M×8b DDR2 SDRAM 芯片的电路框图(N_R 为 14,N_L 为 10,N_B 为 2,M 为 8,页面大小为 1 KB)。

与非同步 DRAM 芯片不同,SDR 和 DDR SDRAM 芯片输入地址(行/列地址 A0 ~ A13,存储体地址 BA0 和 BA1)、命令控制(片选 CS#、行选 RAS#、列选 CAS#和写入控制 WE#)和数据的输入输出(DQ0 ~7)及字节控制 DQM 均需时钟同步。即根据命令控制的电平组合,在时钟的上升沿输入命令,确定内部的工作状态和状态的转换,产生内部的行选和列选时序控制。在时钟的上升沿输入相应的行或列地址。由于同一行的存储单元在该行选中时,可同时读出到列读出放大器或由列写入数据寄存(驱动)器写入,所以可实现突发(成组)传送的读出或写入操作。即在输入第一个列地址后,后续列地址不需由芯片外输入,而由列地址计数器在输入的第一个列地址的基础上,按加电后设置的突发传送地址生成规则和突发传送长度产生。后续地址存储单元的读出时间,只是输出寄存器输出相对于时钟信号的延迟。

DDR SDRAM 芯片在 SDRAM 基础上每数据字节增加了双向选择信号 DQS。数据输入输出以 DQS 的上升下降边为参考。写入时 DQS 为输入,读出时 DQS 为输出。在相同的时钟频率下,其数据速率是 SDRAM 芯片的数据速率的 2 倍。提高了时钟频率和数据传输率后,需特别关注数据、DQS、时钟、地址和控制命令等高频信号传输的完整性。为此 DDR SDRAM 芯片时钟输入改为差分输入,以减小外部干扰对时钟抖动的影响。芯片内增加了控制 DQS 信号的延迟锁定环(DLL)电路,以保证正确的源同步。DDR2 SDRAM 芯片将 DQS 信号改为差分信号,并采用芯片内的终端匹配电阻(ODT)及数据和 DQS 驱动输出电阻可调(OCD)技术。DDR3 SDRAM 芯片又进一步发展了 ODT 的校准技术。以保证在很窄的选通数据窗口内,能正确地接收和发送数据。

DRAM 的刷新 再生地址计数器可提供执行刷新操作时的行地址。非同步 DRAM 芯片有三种刷新方式:仅 RAS 刷新方式,CAS 在前的刷新方式(有控制内部刷新地址计数器的计数操作),隐含方式(此方式类似于 CAS 在前的刷新方式,但与读写操

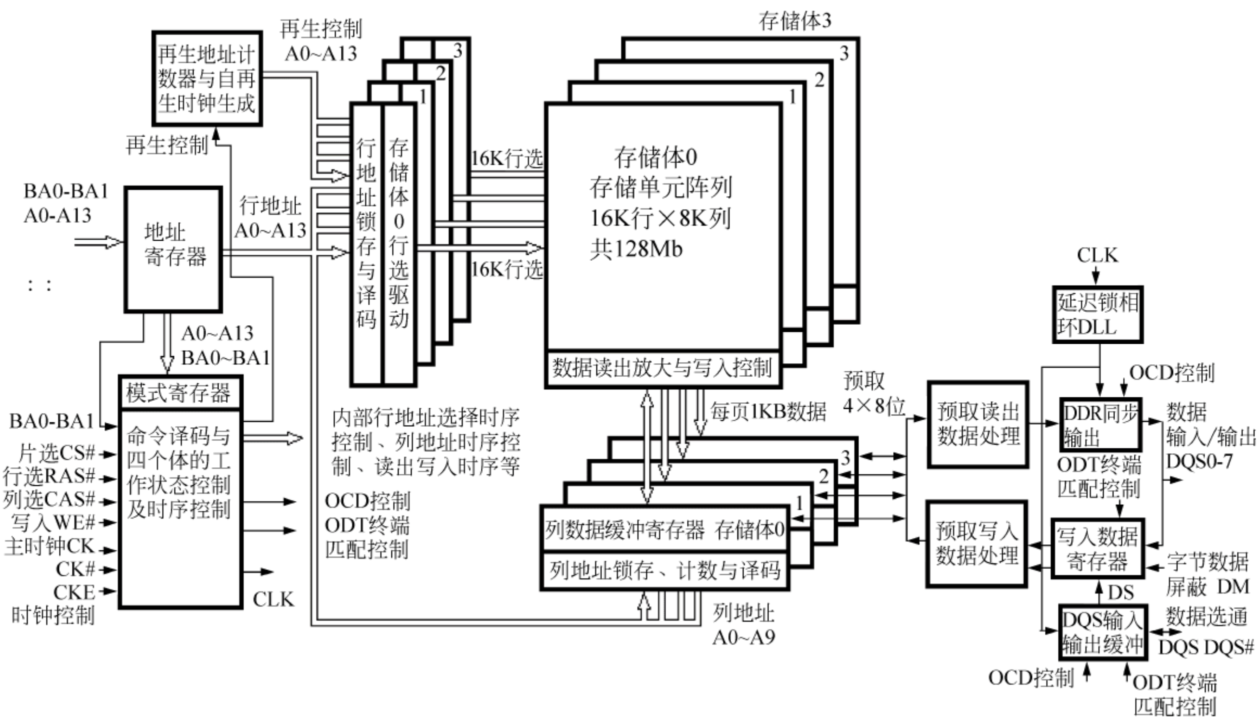


图 2 512 Mb (4×16 M×8 b) DDR2 SDRAM 芯片框图

作合并进行,也可用内部刷新地址计数器确定刷新地址)。SDRAM 提供两种刷新方式:自动刷新和自刷新。自动刷新用于有时钟输入的场合,通过输入自动刷新命令(CBR),各个存储体同时刷新内部再生地址计数器确定的一行存储单元,并使计数器加1。在输入 CBR 命令前,必须先执行所有存储体预充电命令。自刷新在系统待机无时钟输入时使用,先执行进入自刷新命令(REFS-EN)后,再进入待机状态;开始内部刷新地址计数器计数和刷新操作,刷新控制电路按一定的时间间隔提供刷新控制信号,以保证各行相邻两次刷新的间隔时间在规定的范围内,在低功耗待机时,所存数据不丢失。系统从新工作时必须先执行退出自刷新命令(REFS-EX),退出自刷新状态,由控制器发 CBR 命令。

刷新间隔时间通常以需刷新的行数(即刷新周期数)和每行连续两次刷新的最大间隔时间表示。例如 256 Mb SDRAM 芯片可以集中刷新,每隔 64 ms,连续对 8192 行刷新。也可分散刷新,即每隔 7.8 μs 刷新一行,每行连续两次的刷新间隔时间不超过 64 ms。

存取速度　这是表征 DRAM 芯片性能的一项指标。对非同步 DRAM 芯片,通常用行地址读出时间 t_{RAC}表示,一般为 60 ns、50 ns、45 ns。实际使用时,列地址读出时间 t_{CAC} 也很重要,它是由 CAS 下降边到读出数据有效的时间,与 t_{RAC} 对应的 t_{CAC} 为 15 ns、13 ns、12 ns。对 SDRAM 芯片通常用时钟频率表示。DDR 类 SDRAM 芯片用数据传输速率和相应的时钟频率下所用的 CL-t_{RCD}-t_{RP} 周期数表示,如表 1 所示。表 1 为不同类型 DRAM 典型芯片的存取速度、容量和电源电压值。由表 1 可看出 SDRAM 和 DDR 芯片加快了连续访问相同行地址的后续列地址存储单元的访问时间,但行选读出时间只有小的改进。

表 1　典型 DRAM 芯片的存取速度、容量和电源电压

	FPM-50 DRAM	EDO-45 DRAM	SDR SDRAM	DDR-333 SDRAM	DDR 2-668 5-5-5 SDRAM	DDR 3-1333 8-8-8 SDRAM
芯片典型容量例	64 Mb	64 Mb	512 Mb	512 Mb	1 Gb	2 Gb
电源电压	3.3 V	3.3 V	3.3/2.5 V	2.5 V	1.8 V	1.5/1.35 V
时钟周期 t_{CK}/频率 ns/MHz	无时钟输入	无时钟输入	7.5 ns/133 MHz	6 ns/167 MHz	3 ns/333 MHz	1.5 ns/667 MHz
页面方式数据速率(每个引脚)	28.6 Mb/s	58.8 Mb/s	133 Mb/s	333 Mb/s	667 Mb/s	1333 Mb/s
行地址读出时间 t_{RAC}	50 ns	45 ns	42.4 ns $5t_{CK}$ +5.4 ns	33 ns $5.5t_{CK}$	30 ns $10t_{CK}$	24 ns $16t_{CK}$
行列地址输入间的延迟 t_{RCD}	20 ns	11 ns	20 ns $3t_{CK}$	18 ns $3t_{CK}$	15 ns $5t_{CK}$	12 ns $8t_{CK}$
t_{RC}行选最小周期时间/时钟周期数	90 ns	74 ns	65 ns/$9t_{CK}$	60 ns/$10t_{CK}$	54 ns/$18t_{CK}$	48 ns/$36t_{CK}$
列数据等待周期 CL/列地址读出时间	/13 ns	/12 ns	$3t_{CK}$ /20.4 ns	$2.5t_{CK}$ /15 ns	$5t_{CK}$ /15 ns	$8t_{CK}$ /12 ns
页面或突发方式数据读出/写入周期	20 ns	17 ns	7.5 ns	3 ns	1.5 ns	0.75 ns
t_{RP}最小预充电时间/周期数	30 ns	25 ns	20 ns/ $3t_{CK}$	18 ns/ $3t_{CK}$	15 ns/ $5t_{CK}$	12 ns/ $8t_{CK}$
产品存储容量种类			64/128/256 Mb	128/256/ 512 Mb,1 Gb	512 Mb,1/2 Gb	1/2/4 Gb
芯片存储体数	1	1	4	4	4,8	8

DRAM 芯片的工作电压向低电压发展,以降低功耗降低芯片温度,保证工作的稳定性。DRAM 芯片用 3.3 V,DDR-333 降到 2.5 V,新的 DDR 3 芯片甚至降到 1.5 V 或 1.35 V。高速和高密度(大容量)工艺本身的发展要求允许的最高工作电压也要相应降低,伴随工艺技术的发展,芯片的存储容量也相应得到提高。

为适应数字系统对存储容量和字长的需要,按字长以各种 DRAM 芯片为基础,设计了标准的单列存储器模组(SIMM)、双列存储器模组(DIMM)和用于笔记本电脑的 SODIMM。这些模组也可简称内存条,以便使用者根据需要选择或扩展容量。由于引脚信号、时序和电源的不同,SDRAM、DDR、DDR 2、DDR 3 有各自的 DIMM、SODIMM 内存条,其尺寸和引脚数是不同的,相互不兼容。

为满足特殊应用的要求,也有以 DRAM 芯片存储单元阵列为内核,输入输出数据速率为 SDR/DDR 的专用 SDRAM 芯片,如图形存储器芯片(参见**视频图形随机存取存储器芯片**)和移动 SDRAM 芯片等。

2011 年 2 Gb(512 M × 4 或 256 M × 8)和 4 Gb(1 G × 4 或 512 M × 8)DDR 3 芯片的数据传输率已达 1.6 Gb/s 和 1.87 Gb/s。采用 30 nm 工艺的 DDR4 芯片和 DIMM 条的样品也已问世,数据传输率达 2.4 Gb/s,电源电压 1.2 V,每 64 位数据通道只支持一条 DIMM,以实现点对点传输。据 JEDEC 国际标准,DDR 4 的数据传输率范围为 2133 ~ 4266 Mb/s。

参考文献

1. Sharma A K. 先进半导体存储器——结构、设计与应用. 曾莹,等译. 北京:电子工业出版社,2005

2. 桑野雅彦. 存储器 IC 的应用技巧. 王庆,译. 北京:科学出版社,2006

3. http://www.samsung.com/Products/Semiconductor

(孙祖希)

duanyu jiegou wenfa

短语结构文法(phrase structure grammar)

形式语言理论中的一种重要文法。一个四元组 $G=(\Sigma,V,S,P)$,其中 Σ 是终结符的有限字母表,V 是非终结符的有限字母表,$S(\in V)$ 是开始符号,P 是生成式的有限非空集,P 中的生成式都为 $\alpha\to\beta$ 的形式,这里 $\alpha\in(\Sigma\cup V)^*V(\Sigma\cup V)^*$,$\beta\in(\Sigma\cup V)^*$。短语结构文法又称为 **0 型文法**。因对 α 和 β 不加任何限制,故也称其为无限制文法。0 型文法生成的语言类与图灵机接受的语言类相同,称为 0 型语言类(常用 $\mathscr{L}_0$ 表示)或递归可枚举语言类(常用 $\mathscr{L}_{re}$ 表示)。

例如,$G=(\{a\},\{[,],A,D,S\},S,P)$,其中:$P=\{S\to[A],[\to[D,D]\to],DA\to AAD,[\to\wedge,]\to\wedge,A\to a\}$,显然,$G$ 是短语结构文法,它所生成的语言 $L(G)=\{a^{2^n}\mid n\geqslant 0\}$ 是 0 型语言。

0 型语言在一些代数运算下的封闭性如表 1 所示,关于判定问题的一些结果如表 2 所示。表中 D 表示可判定,U 表示不可判定,G 表示文法,L 表示语言。

表 1 $\mathscr{L}_0$,$\mathscr{L}_1$ 在代数运算下的封闭性

代数运算 \ 语言类	$\mathscr{L}_0$	$\mathscr{L}_1$
求并	✓	✓
求连结	✓	✓
求闭包	✓	✓
求补	×	?
求交	✓	✓
与正则语言相交	✓	✓
反演	✓	✓
置换	✓	×

注:✓表示封闭;×表示不封闭;? 尚未解决。

表 2 与 $\mathscr{L}_0$,$\mathscr{L}_1$ 有关的判定问题

判定问题 \ 语言类	$\mathscr{L}_0$	$\mathscr{L}_1$
任意字 $x\in L(G)$?	U	D
$L(G_1)\subset L(G_2)$?	U	U
$L(G_1)=L(G_2)$?	U	U
$L(G)=\varnothing$?	U	U
$L(G)$ = 无限集合?	U	U
$L(G)=\Sigma^*$?	U	U

短语结构文法的标准型为:$A\to\xi$,$A\to BC$,$A\to\wedge$,$AB\to CD$,其中 $\xi\in(\Sigma\cup V)$,$A,B,C,D\in V$,$\wedge$ 是空字。

参考文献

Hopcroft J E, Ullman J D. Introduction to automata theory, languages and computation. Boston, MA: Addison-Wesley,1979

(马世骅)

duideng jisuan

对等计算(peer-to-peer computing)

一种区别于传统客户机服务器的计算模型,采用这种计算模型构建的计算系统称作对等计算系统(peer-to-peer,P2P)。P2P 系统中的任何节点都可以同时充

当系统中的服务器和客户端。任何信息通信和交换活动都是在该活动的各参与节点之间直接完成,整个系统中的所有节点均处于同等地位。因此,P2P 系统与其他**互连网络**的主要区别就在于它在大量的终端用户之间进行数据的共享,而不是通过一些中心服务器或者 Web 服务器进行共享。

P2P 系统中的一个核心机制是数据的查找。这种机制依赖于数据以及网络是如何组织的。P2P 系统中的数据查找是以数据为中心,而不是传统网络中那些以主机为中心的算法。P2P 查找使用 P2P 应用层覆盖网络。它是一种节点间的逻辑关系图。P2P 网络中的对象查找、对象存储及管理算法都与覆盖网络结构关系密切。

按照覆盖网络的种类,P2P 系统可以分为结构化和非结构化两种。结构化 P2P 系统采用诸如超立方体结构、网格结构、蝴蝶网络结构以及德布鲁因图结构等来进行节点的组织,它往往采用分布式哈希表(DHT)进行数据对象的存储和定位,其优点是数据对象的精确查找速度快,缺点是面对节点加入或退出系统时,需要进行的处理比较复杂;非结构化 P2P 系统则不使用任何特定的图结构进行节点组织,其优点是结构灵活,较能够忍受许多节点快速加入和退出 P2P 系统的行为,缺点是对象查找和定位困难,往往需要通过低效率的洪泛查找或随机走步方式来完成对象定位。

常见的 P2P 系统

(1) Napster 是最早的热门 P2P 系统之一。它使用一组中心式集群服务器作为中介节点,来存储整个系统中所存储文件的直接索引。客户机首先向中心服务器发出其要查找内容的请求,服务器随后对客户机进行响应,将该服务器上存储的所有用户以及他们所共享的文件发送给客户机,收到服务器响应后,客户机从中选择一个用户资源进行下载,在该客户机以及其所选择的目标用户间的网络连接由服务器来中转。

(2) Gnutella 是一个非结构化的 P2P 系统。它采用的是一种全分布式的架构,每个用户都对其本地内容进行索引,但并没有一个中心式的索引服务器来进行全局内容的索引,而是由所有用户的本地索引一同构成一个全局索引。当用户请求某个文件的时候,其请求在用户间被转发,直至该请求到达存有该内容的用户,最终在两个用户间直接完成文件的共享操作。

(3) Chord 是一个结构化的 P2P 系统构造方法,它使用一个扁平的键值空间将网络节点与数据对象/文件/值进行了映射。通过使用一个一致性哈希函数,节点地址以及数据对象/文件/值被映射为一个普通键值空间的逻辑编号。所有这些映射都应确保所有键值大致均匀地在节点间进行分布,这也确保了当节点加入或离开 P2P 网络的时候,产生大的键值管理额外开销的概率变得很低。

(4) Tapestry 是一个结构化的 P2P 系统构造方法,它提供了有效、可扩展,并与位置无关的路由方法,以便在 Tapestry 节点中定位对象,其设计是由更早的 Plaxton 树演化而来的。Tapestry 的重要改进包括处理因为节点加入退出产生的网络扰动以及动态的对象加入和删除。就像 Chord 一样,节点和对象被赋予一个使用类似 SHA-1 的一致性均匀分布哈希函数将本地名字空间映射到一个普通标识号空间后得到的标识号。

参考文献

Kshemkalyani A D, Singhal M. Distributed computing: principles, algorithms and systems. Cambridge University Press. 2008

(余宏亮)

duideng moshi

对等模式(peer-to-peer mode) 一种不同于传统的客户端/服务器模式的通信模式,能够将任务分布到不同的节点,对等模式的节点具有同等的能力,既可以作为服务器共享资源供网络中其他节点使用,又可以作为客户端请求其他节点的资源。对等模式中网络通信的任何一方可以发起通信会话,节点之间可以直接通信、共享资源、协同工作。对等模式能够不依赖于中心节点而依靠边缘节点自组织以对等协作的方式进行资源发现与共享,具有自组织、自管理、可扩展性好、鲁棒性强以及负载均衡等优点。

对等模式中节点的通信模式与传统的客户端/服务器模式不同,对等模式中的节点作为平等的个体参与到网络中,节点贡献部分自身的资源,比如处理能力、存储空间、网络带宽等,以供其他节点利用,而不需要服务器或者稳定主机的中心调度。传统的客户端/服务器模式中,服务器仅仅作为资源的提供者,客户端仅仅作为资源的消费者,而对等模式的节点在网络中既作为资源的供应者,也作为资源的消费者。

对等模式从 Napster 等文件分享系统出现后开始流行,并在短短几年时间内迅速发展,目前已经成

为互联网上广泛应用的技术之一。**P2P 技术**是对等模式的典型应用,P2P 技术以其分布式资源共享和分布并行传输的特点,为用户提供了更多的存储资源、更高的可用带宽以及更好的服务质量,同时也减少了内容提供商的带宽消耗,节省了内容提供商的费用开销。目前 P2P 技术广泛应用于文件共享、网络视频、网络语音等领域,产生了多种多样的 P2P 应用。从 P2P 应用的实现原理来看,P2P 应用并不是一种高效率的传输模式,因为在 P2P 应用的传输过程中存在很多重复的数据分组,占用了大量的网络带宽,甚至造成网络拥塞,从而降低其他应用的性能,但是 P2P 应用利用多路并行传输带来的快速传输性能却使其他应用难以望其项背。

对等模式虽然得到了广泛的应用,但是仍然面临着许多发展障碍,比如版权问题、网络带宽问题、管理问题和安全问题等,此外对等模式还缺乏统一的标准,采用不同对等模式的网络间的互通存在着障碍。

参考文献

徐恪, 吴建平, 徐明伟. 高等计算机网络——体系结构、协议机制、算法设计与路由器设计. 2 版. 北京: 机械工业出版社, 2009 (徐恪)

duideng moxing

对等模型(peer-to-peer model, P2P) 互联网环境下一种支持各参与节点进行直接通信的**分布式计算模型**。在 P2P 模型中,参与节点的本地资源无须集中于个别中心节点,且各节点通常具有很强的自治性和平等性,可同时作为请求资源的客户方和提供资源的服务方。

P2P 模型的核心是资源定位,即快速找到所需资源的所在节点位置,从而使参与节点之间得以直接进行通信。基于 P2P 模型的资源定位和通信过程大致包括四个基本步骤:①网络构造 作为互联网之上的一种应用层网络,所有参与节点组成覆盖网络,覆盖网络在资源定位时的路由信息由各节点共同维护;②资源发布 各节点将其拥有的资源的位置信息按某种存放策略动态地发布到覆盖网络的相应节点上;③资源定位 需要某种资源的请求节点按某种查询策略从覆盖网络的某个节点上获得所需资源的位置信息;④对等通信 请求节点直接与目标节点通信,访问所需资源。

企业中用得最多且最简单的资源定位策略是覆盖网络中设置一个或多个节点专门用于存放、维护和管理所有资源的位置信息,其他所有节点的资源发布和资源定位均通过访问此类节点完成,采用这一策略的对等模型称为混合型 P2P 模型。其后出现的 P2P 模型的资源定位策略是,每个节点按通信应答时间或其他因素确定其邻居节点,各节点的资源发布和资源定位仅与它的邻居节点直接通信,覆盖网络通过广播机制经多步转发实现资源位置信息的发布和定位,此类覆盖网络没有特定拓扑结构,相应的 P2P 模型称为无结构 P2P 模型。学术界对 P2P 模型也开展了广泛研究,他们提出了一类称为分布式散列表的位置映射机制,各节点利用一个公共的散列函数计算资源位置的散列值,将资源的位置信息映射到覆盖网络相应的节点上,此类 P2P 模型的覆盖网络具有准确严格的拓扑结构,通常称为结构化 P2P 模型。

P2P 模型有利于充分利用网络带宽,并具有很高的可扩展性和良好的容错性,使大规模节点之间的资源共享与信息交换成为可能,目前 P2P 模型在各类基于互联网的信息网络服务系统中得到了广泛应用。

参考文献

陈贵海,李振华. 对等网络:结构、应用与设计. 北京:清华大学出版社,2007 (吴泉源 尹刚)

duideng xiazai

对等下载(peer-to-peer download) 一种基于覆盖网络上的文件共享方式,又称 P2P 文件共享。参与下载的节点通过应用层的逻辑链接构建起一个覆盖网络,文件资源的查找、交换即在该覆盖网络上进行。对等下载最大的特点是,资源分布式地存储于参与下载的各节点而非一个服务器上,一个节点在从别的节点下载文件的同时,也为另一些节点上传其所拥有的文件资源。通过充分利用处于网络边缘节点的软硬件资源,避免了传统 C/S 模式中由服务器造成的瓶颈问题。

1998 年,18 岁的美国大学生肖恩·范宁编写了第一个 P2P 文件共享软件 Napster,为用户提供存储于其他用户机器上的音乐文件的查询与检索服务。Napster 出现后,P2P 文件下载技术迅速发展,各种 P2P 文件下载软件相继出现,逐渐取代了传统的 C/S 模式,成了文件共享的主流方式。

构建一个 P2P 文件下载系统通常有两个方面的内容:参与下载的节点的组织,文件的查询与定位。根据对集中控制依赖程度的不同,P2P 文件下

载可以分为如下三类:集中目录模式,纯 P2P 模式,分层模式。其技术实现各不相同。

集中目录式 P2P 文件下载

在集中目录式 P2P 文件下载系统中,存在一个存储元数据的中央服务器。新加入系统的节点首先要向服务器发出加入请求,登记自己的相关信息(IP、带宽、文件资源等)。需下载文件前,先向服务器发出相应的查询请求,服务器收到查询请求后,在本地完成查找,然后返回拥有相应文件的节点的信息作为查询结果,需下载文件的节点根据查询结果与相应节点建立链接,完成文件传输。

与传统 C/S 模式不同,集中目录式 P2P 文件下载系统中的服务器并不存储文件的实际数据,只为下载节点提供查询、检索服务,文件的下载、上传在各节点间完成。这样便极大地减轻了服务器端负载,降低了系统成本。

纯 P2P 式文件下载

纯 P2P 文件下载方式去掉了对中央服务器的要求。参与下载的节点功能相同,通过相互之间的逻辑链接构成一个覆盖网络,文件及系统相关信息都分布式地存放在这个覆盖网络上。对文件的查询通过信息在覆盖网络上的传递来完成。纯 P2P 文件下载的最大特点在于其回避了由中央服务器造成的单点失效问题。根据覆盖网络的不同,纯 P2P 文件下载又可以分为无结构的纯 P2P 文件下载及有结构的纯 P2P 文件下载。

分层的 P2P 文件下载

分层的 P2P 文件下载与纯 P2P 模式基本相同,区别在于前者引入了部分集中控制的思想。系统选择一些性能较好的节点作为超级节点,超级节点负责对挂靠于它的节点及这些节点的文件信息的管理。在原有覆盖网络的基础上,由超级节点间链接构成另一个高一层的覆盖网络。节点向其挂靠的超级节点发出文件查询请求,这些请求在超级节点间转发,直到找到相应文件。这种分层结构利用超级节点来完成部分中央服务器的功能,在回避了中央服务器的同时提高了查询效率,增强了系统的可扩展性,降低了维护代价。

参考文献

1. 陈贵海, 李振华. 对等网络:结构、应用与设计. 北京: 清华大学出版社, 2007

2. 张春红, 裘晓峰, 等. P2P 技术全面解析. 北京: 人民邮电出版社, 2010

(陈贵海)

duixiang

对象(object)　面向对象系统中运行时刻的基本成分,它是数据和操作(或谓属性和行为)的封装通信单位。数据表示对象的属性状态;操作(或称方法)决定了对象的行为和与其他对象进行通信的接口。数据是对象私有的,只有该对象内的操作才能进行存取。

任何事物均有各自的属性与行为,当考察其某些属性与行为并进行研究时,它便成为有意义的对象。当用面向对象的方法进行计算机模拟时,应区别两种不同含义的对象:**问题对象**和**计算机对象**。前者是指现实世界中存在的实体在问题域中的抽象,后者是指问题对象在计算机系统中的表示。

面向对象语言中的对象指的是计算机对象。对象具有被动和主动两个方面。被动方面指的是其相对静态属性,借以认识对象,并对之归类。主动方面指的是其具有改变静态属性的动态行为。静态属性与动态行为相互影响,属性决定行为,行为改变属性。此外,对象并非孤立存在,彼此之间通过传递消息进行交互。这样,对象可表示成三元组(接口、状态、操作)。动态地看,对象又是通信自动机。

对象内部有两类操作:一类是改变其内部属性的操作,另一类是生成输出的操作。

对面向对象语言中的对象,流行的观点有两种:一是以 **Smalltalk 语言**为代表的广义解释,将对象定义为计算机系统中的所有成分,强调动态系统本身;另一是以 **Eiffel 语言**为代表的狭义解释,将对象定义为计算机系统中用以模拟问题对象的成分,着重考察系统和现实世界间的对应关系。

计算机对象具有 5 个基本特性: ①自治性,指对象具有一定的独立计算能力; ②封闭性,指对象具有信息隐蔽的能力; ③通信性,指对象具有与其他对象通信的能力; ④被动性,指对象的状态转换由外部刺激引发; ⑤暂存性,指对象的动态创建与消亡。

参考文献

1. 徐家福,等. 对象式程序设计语言. 南京: 南京大学出版社,1992

2. Special Issue on object-oriented design. CACM,1990,33(9)

(张家重　王志坚)

duixiang cunchu xitong

对象存储系统(object-based storage system)　由多个基于对象的存储设备、元数据服务器、客户

端、互连网络组成的存储系统。该系统采用 SCSI（参见**外存储设备接口**）基于对象的设备命令集，结合了**附网存储**（NAS）中的基于文件的访问接口以及**存储区域网**（SAN）中基于块的访问接口的优点，提供比其他两种接口更为丰富的语义。对象存储设备利用其上富裕的计算资源承担了原本由服务器负责的**存储管理**功能，自动管理其上的数据分布，具有一定的智能性。对象存储系统是基于对象的存储设备构建的存储系统，特点是将数据通路（数据读或写）和控制通路（元数据）分离，实现对象存储设备和客户端的数据高速并行访问。

对象存储研究源于卡内基梅隆大学（Carnegie Mellon University）并行数据实验室（parallel data lab，PDL）1994 年的 NASD（network-attached security disks）项目。此后 1999 年成立了全球网络存储工业协会（SNIA）的对象存储设备（object based storage device，OSD）工作组，发布了美国国家标准学会（ANSI）的 X3 T10 标准。2004 年，ANSI 推出了 OSD 1.0 版本规范，定义了对象存储设备的通信协议。2008 年 SNIA 技术工作组发布 OSD 规范 2.0 版本。

对象是数据和属性的结合体，由对象的身份标识号码（identity，ID）来唯一标识，属性可以根据需求设置。对象存储设备存放对象，并控制着从对象到物理介质的映射图。对象存储设备同时也跟踪作为对象属性的本地元数据信息。对象存储设备内对象通常分为四类：根对象（root object），分区对象（partition object），集合对象（collection object）和用户对象（user object）。

对象存储系统的架构如图 1 所示，主要包括对象存储设备、元数据服务器、客户端三部分，并通过高速网络互联。

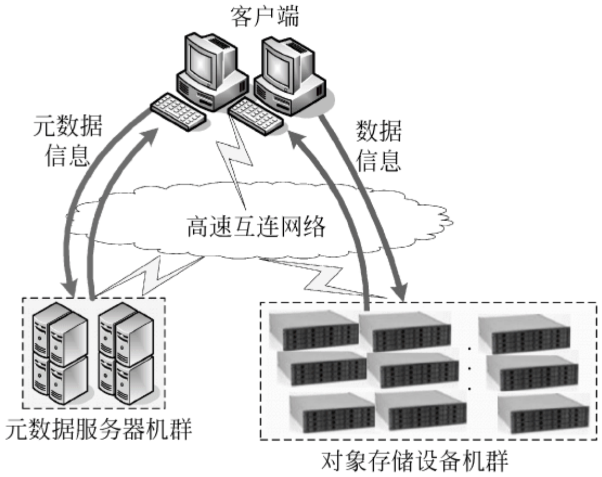


图 1　对象存储系统结构示意图

对象存储设备（object based storage device，OSD）　拥有自己独立的中央处理器（CPU）、内存、网络和磁盘系统，具有一定的智能性，与块设备的不同不在于存储介质，而在于所提供的访问接口为对象接口。OSD 的主要功能包括数据存储、数据安全访问等。OSD 提供的三个主要功能是：

（1）数据存储　OSD 管理对象数据，并将它们放置在标准的磁盘系统上，OSD 不提供块接口访问方式，客户端请求数据时用对象 ID、偏移进行数据读写。

（2）智能分布　OSD 用其自身的 CPU 和内存优化数据分布，并支持数据的预取、并行操作、自动迁移等。由于 OSD 可以智能地支持对象的预取，根据热点访问生成副本和对数据重新布局，从而可以优化存取性能。

（3）每个对象元数据的管理　OSD 管理存储在其上对象的元数据（参见**多媒体数据库**），该元数据与传统的 inode 元数据相似，通常包括对象的数据块和对象的长度。而在传统的 NAS 系统中，这些元数据是由元数据服务器集中维护，而对象存储架构将系统中主要的元数据管理工作由 OSD 来完成，减少了元数据服务器负载。

元数据服务器（metadata server，MDS）　控制客户端与对象存储设备的交互，主要提供以下几个功能：

（1）对象存储访问　MDS 构造、管理描述每个文件分布的视图，根据请求判定是否授权客户端直接访问对象的能力。对象存储设备在接收到每个请求时将先验证该能力，然后才可以访问。

（2）文件和目录访问管理　MDS 在存储系统上构建一个文件结构，包括限额控制、目录和文件的创建和删除、访问控制等，向用户提供统一的文件系统视图。

（3）客户端缓存一致性　为了提高客户端性能，在对象存储系统设计时通常支持客户端的缓存。由于系统存在多客户端，带来了**高速缓冲存储器**（cache）一致性问题，MDS 支持基于客户端的文件 cache，当 cache 内的文件发生改变时，将通知客户端刷新 cache，从而防止 cache 不一致引发的问题。

对象存储系统的客户端　为了有效支持并行访问 OSD 上的对象，需要在计算节点安装对象存储系统的客户端，通常提供 POSIX 文件系统接口，允许应用程序像执行标准的文件系统操作一样访问对象存储系统。

对象存储系统已广泛应用于高性能计算领域。

国际上具有代表性的对象存储系统有 Panasas 和 Lustre。Panasas 将基于对象的分布式文件系统与专用硬件结合，充分利用其单一的全局名字消除烦琐的操作任务，降低了管理成本，同时通过并行路径优化技术使其总体性能表现优异。Lustre 是开源的对象存储技术，由美国能源部（Department of Energy）开发，惠普公司（HP）提供商业支持。截至 2011 年 6 月，在全球 Top10 超级计算机中有 8 套、Top 100 中有 70% 都采用了 Lustre 对象存储系统。因此从需求及技术先进性看，对象存储系统具有良好的发展趋势。

参考文献

1. 张江陵，冯丹. 海量信息存储. 北京：科学出版社，2003

2. Mesnier M, Ganger G R, Riedel E. Object based storage. IEEE Communications Magazine, 2003, 41(8): 84-90

3. Information technology-SCSI Object-Based Storage Device Commands-2(OSD-2). INCITS T10 Working Draft. http://www.t10.org/drafts.htm

（王芳　冯丹）

duixiang-guanxi shujuku

对象-关系数据库（object-relation database）

支持对象特征的**关系数据库**。

对象-关系数据库基于对象-关系数据模型，该模型通过提供处理复杂对象的数据类型扩充关系模型。

对象-关系型数据库是面向对象技术与传统关系型数据库技术相结合而形成的数据库系统。也可以说是一种扩展关系数据库，使它具有一定面向对象数据库特征。这种数据库系统既保留了传统关系数据库对结构化常规数据的高效访问、安全、事务管理等优点，又扩充了支持多媒体和用户自定义数据类型的灵活性。20 世纪 80 年代中期，为了满足新一代数据库应用的需求，**面向对象数据库**系统的研究与开发应运而生。在面向对象技术与数据库技术相结合的过程中，主要是沿着两条技术路线展开的。一条是建立纯粹的**面向对象数据库管理系统**（OODBMS），支持面向对象数据模型。另一条是以关系数据库和结构查询语言（SQL）为基础加以扩展，增加面向对象特性，建立**对象-关系数据库管理系统**（ORDBMS）。

面向对象数据库系统能较好地满足一些特定应用领域（如**计算机辅助设计**）的需求。但是，纯粹的面向对象数据库系统不支持结构化查询语言（SQL），在通用性方面丢失了关系数据库的优势。1990 年高级 DBMS 功能委员会发表了"第三代数据库系统宣言"，提出了面向对象数据库系统必须满足的两个条件：①支持核心的面向对象数据模型的主要特征；②支持传统数据库系统所有的数据库特征。也就是说，第三代数据库系统必须保持传统数据库系统的非过程化数据存取方式和数据独立性，应该继承数据库系统已有的技术，不仅能很好地支持对象管理和规则管理，而且能更好地支持原有的数据管理。对象-关系数据库系统就是按照这样的目标，将面向对象技术与关系数据库技术结合起来。

对象-关系数据库除了具有关系数据库的各种特点外，还具有以下特点：

（1）扩充数据类型　关系数据库只支持某一固定的类型集，不能依据某一应用所需的特定数据类型来扩展其类型集。对象-关系数据库系统允许用户在关系数据库系统中扩充数据类型，即允许用户根据应用需求自己定义数据类型、函数和操作符。

（2）支持复杂对象　能在 SQL 中支持复杂对象。复杂对象是指由多种基本数据类型或用户自定义的数据类型构成的对象。

（3）支持继承概念　能支持子类、超类概念，支持继承概念，包括属性数据的继承和函数及过程的继承；支持单继承与多重继承；支持函数的一名多用（重载）或操作的一名多用（重载）。

（4）提供强大而通用的规则系统　规则在数据库系统及其应用中十分重要，在传统 RDBMS 中用触发器来保证数据库数据的完整性。触发器可以看成规则的一种形式。对象-关系数据库系统要支持的规则系统更加通用，更加灵活，并且与其他的对象-关系能力集成为一体，例如规则中的事件和动作可以是任意的 SQL 语句，可以使用用户自定义的函数，规则能够被继承等。这就大大增强了对象-关系数据库的功能，使之具有主动数据库和知识库的特性。

由于对象-关系数据库系统既能适应新应用领域的需求，也能继续满足传统数据库应用深化发展的需要，目前几乎所有的关系数据库厂商，都在不同程度上扩展了关系模型，推出了对象-关系数据库管理系统产品。对象-关系数据库系统获得了快速发

展。结构查询语言(SQL)国际标准 SQL3 中也增加了对象-关系数据库管理系统(ORDBMS)的特征。因此,对象-关系数据库系统的应用日趋广泛。

参考文献

1. 王珊,萨师煊. 数据库系统概论. 4 版. 北京:高等教育出版社,2006

2. Stonebraker M M, Moore D. Object-relational DBMSs: the next great wave. Morgan Kaufmann Publisher, Inc., 1996(杨冬青,唐世渭,裴芳,等译. 对象-关系数据库管理系统——下一个浪潮. 北京:北京大学出版社,1997) (李建中 王珊 邹兆年)

duixiang qingqiu daili

对象请求代理(object request broker) 在分**布计算环境**下,一种通过对象代理支持分布对象之间透明访问的机制。

请求服务的对象称为客户对象,提供服务的对象称为服务对象。实体服务对象指实际上实现服务的对象,它与客户对象是分布的,且通常不在同一台机器上。由于传统的**面向对象的方法**不能实现跨机器的对象间访问,于是,出现了实际服务对象的代理,它设置在客户对象的同一台机器上,并全权负责与服务对象间的联系,客户对象只需向本地的服务对象代理发出访问请求,通过该代理的中介,便可以实现从客户对象到异地的实体服务对象的访问。对象请求代理机制的任务就是要通过该代理在请求对象与实际服务对象(服务实现)之间架起一座互连互通互操作的桥梁,支持分布对象之间的透明访问。主要工作包括:基于包交换等网络传输层的通信,传递客户对象请求,定位服务对象,并调用相应的服务方法,返回调用结果等。

对象请求代理作为分布式计算的一种重要的使能技术,其研究内容主要包括服务对象在客户方的代理机制、对象之间请求传递的通信机制、服务对象定位机制以及服务方法调用机制等,如图 1 所示。服务对象在客户方的代理机制负责在客户方创建服务对象在本地的代理,并接收客户对象的请求;对象之间请求传递的通信机制负责在客户对象和服务对象之间传递服务请求以及请求结果;服务对象定位机制根据客户请求在服务器内查找和定位服务对象;服务方法调用机制将客户的服务请求调度到服务对象上,根据请求内容激活其所请求的服务方法,并通过对象之间请求传递的通信机制将结果返回给客户对象。

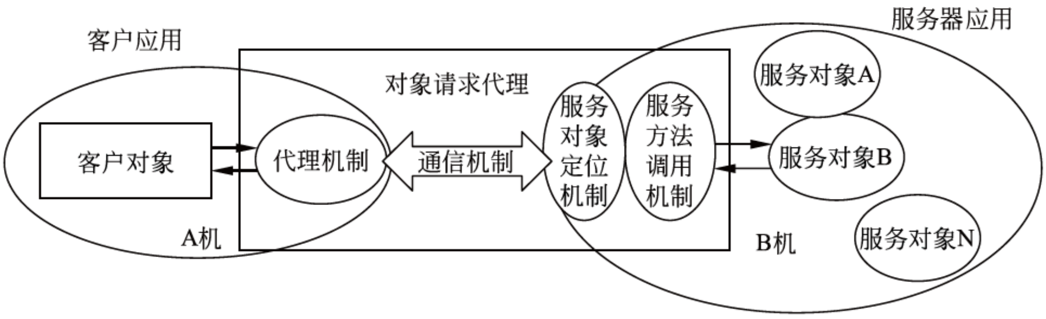


图 1 对象请求代理结构

对象请求代理的主要特征在于提供了客户对象和服务对象之间的交互透明性。具体而言,对象请求代理可以使分布计算环境中的以下元素对客户对象透明:①对象位置透明 客户方无须了解服务对象的具体位置,服务对象可能处于不同机器的不同进程、同一机器的不同进程甚至同一机器的同一进程。②对象实现透明 客户方不需要了解服务对象具体如何实现的,例如其所采用的程序语言等。③对象执行状态透明 即使服务对象尚未被加载或激活,对象请求代理可以透明激活服务对象。④对象通信机制透明 底层的网络通信对客户方和服务对象都是透明的。

对象请求代理的这些特征使得程序员可以专注于业务功能,而不是分布式系统的某些低层传输问题,从而简化了分布式应用软件的开发。对象请求代理是由对象管理组织 OMG 在其制订的分布对象计算规范 CORBA 中提出的,基于对象代理机制的**面向对象中间件**已广泛应用于电信、金融、交通、航空、海事、国防以及信息安全等领域。

参考文献

OMG. Common Object Request Broker Architecture Specification. http://www.omg.org/technology/documents/corba_spec_catalog.htm (史殿习 丁博)

duochuliji xitong zongxian

多处理机系统总线(multiprocessor system bus) 使多个处理机模块、存储器模块、输入输出部件以及网络接口等部件(或多个由中央处理器、局部存储器、输入输出部件所组成的计算机模块)实现互连的公共通路。这是组成多处理机系统的一种最简单最直接的结构形式。在多处理机系统总线中,欲传送的信息以分时或多路转换方式在系统的主控部件和受控部件之间传送。主控部件主要是处理机或计算机模块,受控部件可以是存储器模块或是I/O部件,也可以是处理机或计算机模块。

多处理机系统总线通常由在各种功能板上的系统总线、存储器总线和局部总线所组成,在中央处理器(CPU)、存储器、I/O或网络接口板上都有局部总线。在存储器板上的局部总线称为**存储器总线**。典型的**I/O总线**是SCSI(小型计算机系统接口)或I/O通道,用来连接本地的磁盘或其他外部设备。设计多处理机系统总线的关键技术包括:总线仲裁、中断处理、协议转换、快速同步、**高速缓冲存储器一致性**协议、分离式的事务处理、总线桥接以及层次扩展等。

多处理机系统总线一般限制在较小的机架内,因此,能由系统总线连接的处理器数就十分有限,为此常用层次式总线结构来提高其可扩展性。

总线互连方式的优点是:简单、价廉、易实现和增减模块灵活方便。因此,早期的多处理机系统几乎全都采用总线互连,至今仍有不少多处理机系统部分使用总线结构。

总线互连的主要缺点是:一是可靠性差,易成为系统的单故障点,一旦总线失效将导致整个系统失效;二是带宽较窄,易成为系统的性能瓶颈。因为系统总线是共享介质网络,其带宽将为连接到总线的部件、模块和处理机所共享,因此可连接到总线的部件、模块和处理机的数量有限,过多的总线负载将导致主控器争用总线的加剧和信息传送时间的加长,从而使整个系统性能低下。多处理机系统总线通常以**事务方式**工作。简单的事务工作方式为连接式的,即在单一的事务操作中只能实现一个主控部件请求和一个受控部件的响应。复杂的事务工作方式为分离式的,它将一个总线事务操作分成请求阶段(包括仲裁、请求和错误检查)以及应答阶段(包括完成、响应和传送数据),它允许总线先执行一个事务的请求阶段,然后在执行该事务的应答阶段之前执行另一个事务的请求阶段,从而以流水方式使用总线。这种工作方式可明显提高总线的带宽,但同时也将使一个事务操作的时延有所增加。

解决上述问题,通常采用以下两种方法:一是改善总线传输特性,用优质电缆组成总线,以提高总线传输速率;二是改变总线结构,增加处理机间的可能通信路径,减少竞争。具体的实现方案有3种:①重复设置多套总线,增加处理机之间可能利用的通信路径,从而成倍地加宽其有效传输频带;②按性质分别设置处理器总线、存储器总线、I/O总线等,再利用总线耦合器把它们连在一起;③采用分级总线结构。尽管总线结构在扩大系统规模上似乎有较大的局限性,但在新近发展的不少多处理机系统中,它仍然是获得广泛应用的一种主要结构形式,其原因一个是因为总线结构自身的改进;另一个则是受微处理器分布处理系统的影响。

多处理机系统总线通常由计算机厂商自行设计,因而是专用的,以获取高的带宽性能,如Sun公司的Gigaplane多处理机总线、SGI公司的POWERpath-2多处理机总线等。多处理机系统总线的发展趋向是标准化(参见**总线标准**)。已有一些总线标准支持多处理机系统。常用的有VME总线、Multibus-Ⅱ总线和Futurebus等。VME总线以异步方式传送信息,Multibus-Ⅱ总线则以同步方式传送信息。用总线结构互连多处理机时,应尽量采用支持多处理机系统的总线标准,如因需要必须采用专用总线时,也应通过总线适配器将它与标准总线相连。唯此各种按总线标准设计的外围设备才可方便地与多处理机系统相连。

多处理机系统总线的工作步骤为:首先,由各个需要使用总线的处理机或计算机模块争用总线使用权;其次,被选中的主控器和受控器与总线相连;最后,信息按一定规约通过总线在一对主控器和受控器间进行传送,必要时也可由一个主控器以广播方式向多个受控器发送。为了解决多个潜在主控器同时争用总线的问题,需由系统总线仲裁器进行裁决,以确定哪个请求源可使用系统总线。

在多处理机系统中,当有1个以上的处理机或计算机模块要求使用系统总线而导致争用时,由**系统总线仲裁器**来裁决该由哪个处理机或哪个计算机模块使用系统总线。系统总线仲裁器一般由仲裁算法和相应的硬件构成。仲裁算法性能的优劣对系统性能有较大影响。

常用的仲裁算法有:

(1) 静态优先级算法　它为每个连到总线上的

处理机(或计算机模块)分配一个唯一的固定优先级。当多个处理机同时请求使用系统总线时,仲裁器使优先级最高的申请者使用总线。常用菊花链方式来确定优先级,越靠近仲裁器的处理机的优先级越高。这种算法的优点是简单、易实现,缺点是优先级低的处理机很少有机会使用总线。

(2) 均等算法　它通常以轮询方式将总线按固定长短的时间片依次供各处理机使用。常用于同步总线。优点是算法较简单且能保证各处理机有均等机会使用总线。缺点是平均等待时间较长。此外,若轮到的处理机不用总线时,将造成总线带宽的浪费。

(3) 动态优先算法　它根据总线使用情况和相应规则,动态地改变连接到总线上的处理机的优先级。如近期最少使用算法,将最高的优先级分给在最长时间间隔内未使用总线的处理机。又如循环菊花链算法,根据离最后一次使用总线的处理机所处位置远近来分配优先级。距离越近的处理机,它的优先级越高。动态优先算法的优点是兼顾了前两种算法的优缺点既有较小的平均等待时间而又可使系统中各处理机有更均等的机会使用总线,缺点是控制逻辑较复杂。

(4) 先来先服务算法　它不按优先级选择申请者,因而有较好的均等性,但实现较困难。主要作为一种衡量其他算法优劣的标准。

上述各种仲裁算法,可用集中式或分布式结构实现。集中式结构由一个仲裁器统一实现仲裁算法,常用轮流查询或独立请求和准用等硬件机构实现。分布式结构则将仲裁硬件分布到各个处理机中,分配给每个处理机一个唯一的优先号,欲请求使用总线的处理机将自己的优先号由各自的分布仲裁器送到共享的请求有效线上进行逻辑"或"操作,形成一个合成优先号。然后再由分布仲裁器将各处理机优先号与此合成优先号相比较,优先号小于此合成优先号的处理机将自动撤销请求,获得总线使用权的将是具有最高优先号的处理机。分布式仲裁结构的主要优点是具有较高可靠性。系统总线仲裁器的工作过程如下:首先通过请求线接收各处理机发来的使用总线请求;然后由仲裁器按照仲裁算法加以裁决并向选中的处理机在总线准用线上发出总线有效信号;最后由被选中的处理机通过总线忙控制线,向其他处理机表明总线已被占用。主控处理机使用总线传送信息后便撤销总线忙信号,从而使仲裁器可再去响应和选择其他处理机对总线的请求。

标准总线都具有仲裁结构,如 VME 总线仲裁器采用集中式结构,而 Multibus Ⅱ 和 Futurebus 等总线仲裁器则采用分布式结构。这些总线一般都支持优先和均等混合仲裁算法,以适应多处理机系统的需要。对外围部件使用优先仲裁算法,而对其他处理机则使用均等仲裁算法,以使各处理机有较均等的机会使用系统总线。

参考文献

1. 黄铠,徐志伟. 可扩展并行计算:技术、结构与编程. 陆鑫达,等译. 北京:机械工业出版社,2000

2. Tanenbaum A S. Structured computer organization. 3rd ed. New Jersey: Prentice Hall, 1990

(陆鑫达)

duodao chengxu sheji

多道程序设计(multiprogramming)　系统同时接纳多个程序进入内存让它们在操作系统控制下交迭或夹插执行的方式。

20 世纪 60 年代初计算机系统中的处理器比外部设备操作的速度要快得多,同时外部设备也能独立地和各种处理器并行地工作。如果仍和过去一样每次只接纳一道程序来执行,那么计算机系统的效率就很低。因为当一道程序启动外部设备后,在外部设备操作结束前程序往往不能继续执行下去而处于等待状态时,尽管处理器可以工作,但没有工作可做。如果系统让若干道程序同时进入内存,当一道程序因为启动外部设备或其他原因暂时不能继续执行而处于等待时,系统可让另一道等待执行的程序来运行。这样做显然可以提高系统的效率。此外,从分时系统的角度看,由于多个用户同时分享计算机系统,如果让用户程序一道道地串行执行,那么用户在终端就可能等待很长的时间,也就不能实现分时的要求了。

实现多道程序设计必须妥善地解决以下三个问题:存储浮动与程序保护,处理器管理和调度以及系统资源的管理。

由于编制程序时无法知道该程序与其他哪些程序同时进入内存,当然也不知道它存放在内存何处,因此,程序中的地址必须是相对的,它可存于内存任何一个区域。此外,由于若干道程序同时存于内存,系统应有相应的保护措施保证各道程序之间互不干扰,特别地,系统必须防止一道程序的出错而影响另一道程序的执行(参见**存储管理程序**)。

处理器调度是实现多道程序设计的一项关键技术。当运行着的一道程序由于启动外部设备或其他原因而暂时不能执行下去时,操作系统的处理器管理就从等待运行的程序中选择一道执行(参见**处理器管理程序**)。

在多道程序设计系统中,系统资源为若干道程序共享,因此**操作系统**必须对共享的系统资源进行管理和调度。操作系统这项功能的主要内容是:分配和去配系统资源,协助使用系统共享资源,例如控制打印机进行打印输出,打开一个文件供用户使用等(参见**文件管理程序**和**输入输出管理程序**)。为了提高系统效率,防止**死锁**发生,对资源管理和调度要设计好调度算法。

在多道程序设计概念的基础上形成了进程的概念。由进程组成的并发程序可以包括多道程序的概念,多道程序概念就包含于多进程概念之中了。

(孙钟秀)

duodian chukong

多点触控(multi-touch) 在一个**触屏**上能同时接受来自屏幕上两点或多点触摸信号的**人机交互技术**。也称多重触控、多点触摸。

多点触控技术包括两个方面,一是同时采集多点信号,二是对每个信号的意义进行判断,也就是所谓的**手势识别**。例如判别运动的方向、起点和终点,图像的平移、滚动、旋转等。用户可通过双手以单击、双击、平移、按压、滚动以及旋转等不同手势触摸屏幕,实现操作。多点触控技术研究始于 1982 年。2007 年 1 月,苹果公司推出了第一款采用多点触控技术的手机 iPhone。经过几年来的发展,现在已有多种采用多点触控技术的产品进入市场。在手机和平板计算机上得到广泛应用。

多点触控采用的技术有:

(1)LLP(laser light plane)技术 运用红外激光设备把红外线投影到屏幕上,当屏幕上有阻挡时,红外线便会反射,而屏幕下的接收器件捕捉到反射方向。通过系统分析,作出位置和动作的判断。

(2)FTIR(frustrated total internal reflection)技术 在屏幕的夹层中加入发光二极管(LED)光线,当按下时,使夹层的光线产生不同的反射效果,从接收到光线变化而得出施力点位置。

(3)互电容检测技术 在单点触屏采用的是自电容检测,即检测每个单元的寄生电容的变化。在多点触屏中,检测的是互电容也就是行列间耦合电容的变化。有手指存在时互电容会减小,就可以准确判断每一个触摸点位置。

多点触控技术,在许多地方取代了键盘、鼠标,使人机交互更简便、更人性化。

(林兼)

duodian chumo jiaohu

多点触摸交互(multi-touch interaction) 用户通过两个或两个以上手指直接触摸显示设备的表面来进行人机交互的技术(见图 1)。该技术无须用户佩戴额外的传感器,具有自然、输入带宽高的优点。多点触摸交互技术已经广泛地应用到手机、平板电脑等设备上,并且正逐步应用到个人电脑、墙面和桌面交互系统中。



图 1

多点触摸交互包括多触点感知技术和多点触摸软件界面。多触点感知技术的主流实现方式分为基于表面电学传感器的(如电容屏、电阻屏等)和基于光学传感器的(如红外对射框、背部红外散射等)。前者的识别精度相对较高,但其中的电阻屏由于仅能支持单点触摸而逐渐被淘汰,电容屏虽然支持多触点,但支持的触点数有限,一般用于小尺寸的设备,比如手机、平板电脑和笔记本;基于光学传感器的多触点识别技术精度相对低,支持点数多,适用于大尺寸交互表面。在大尺寸交互系统中,还有一类基于对射光线阵列框的触摸感知技术,精度和点数受限制,在水平观察的桌面交互中,基于红外激光的还存在对人眼的安全隐患。

与现有的指点设备相比(比如**鼠标**),多点触摸允许多点并行输入以提高交互效率,但同时它又受到"肥手指"问题的困扰,无法准确选取小尺寸的交互元素。这些特点使得传统的窗口界面系统(GUI)不再适应多点触摸输入方式。目前,工业界已经有

不少对多点触摸交互进行优化的操作系统,例如 iOS、Android 和 Microsoft Surface,学术界也研制出了一些用于研究多点触摸交互的开发工具,例如 DiamondSpin 和 uTableSDK。它们都采用了适合手指触控的界面元素,不仅支持多点触摸,还支持手势操作。

多点触摸技术还处于快速发展阶段。随着传感器技术的发展和交互软件的日渐成熟,多点触摸有望取代鼠标等传统指点设备成为新一代的人机交互方式。

参考文献

1. Buxton B. Multitouch overview. March 21st, 2011

2. NUI Group. Multitouch technologies. Version 1.0. 2009

(史元春)

duofashe jiegou

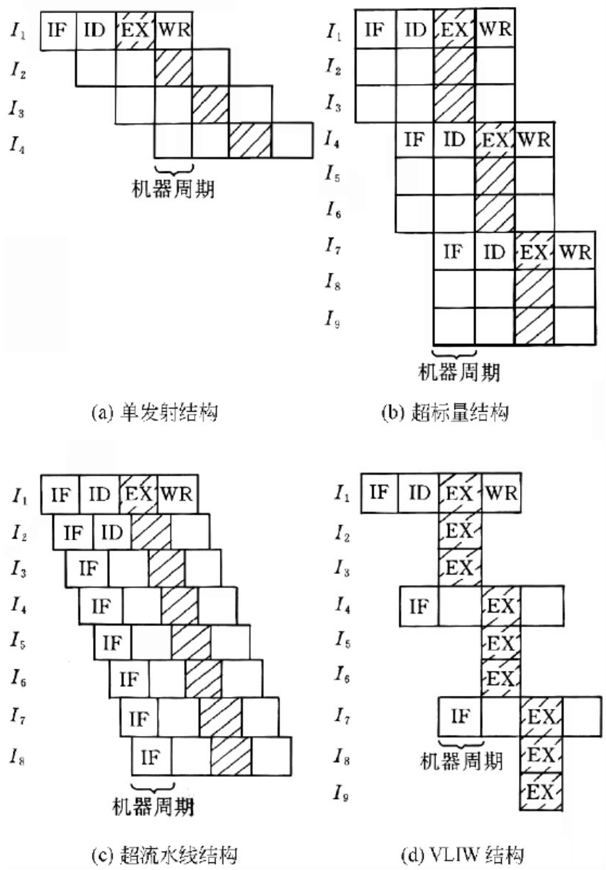
多发射结构(multiple issue architecture)

在 1 个机器周期内可发射多条指令给执行部件以提高指令执行并行性的**处理机体系结构**。

在先进的处理机体系结构中,一般采用流水线结构(参见**计算机流水线**)以提高指令执行的并行性。自**精简指令集计算机**(RISC)推广以后,各种处理机都力图在流水线的基础上把指令系统中的基本指令在 1 个机器周期(参见**指令周期**)执行完毕。这时,普遍采用 CPI(执行每条指令的平均周期数)来衡量处理机的指令执行并行性。由于指令系统中除了基本指令以外,还有一些较复杂的指令,它们不可能在 1 个机器周期内执行完毕。因此,在每个机器周期内只发射 1 条指令的处理机体系结构中(这种结构称为**单发射结构**),CPI 的平均值只会大于 1。在设计优良的 RISC 单发射结构的处理机中,CPI 值可以达到 1.1~1.2。

为了进一步提高处理机指令执行的并行性,即在相同的时钟频率下进一步提高处理机的执行速度,必须减少 CPI。要使 CPI 值小于 1,则必须在流水线的基础上,在每个机器周期内发射多条指令给执行部件,这就是多发射体系结构。

单发射和多发射结构处理机在流水线基础上的指令执行可以用图 1 表示。其中图 1(a)是单发射结构处理机执行指令的流水线情况,流水线分 4 站:IF 为取指令站,ID 为指令译码站,EX 为执行指令站,WR 为写回结果站。图 1(b)、图 1(c)和图 1(d)分别是多发射结构的 3 种主要类型,即**超标量结构**、**超流水线结构**以及**超长指令字**(VLIW)**结构**。超标量结构是在每个机器周期内发射多条指令(图中为 3 条指令)。超流水线结构的指令执行流水线是把 1 个机器周期再细分为多个子周期,每个子周期发出 1 条指令,所以 1 个机器周期可发射多条指令。超长指令字结构(VLIW)是在取指令级 IF 内取出 1 条很长的指令字,这字中可以包含多条指令,如 8 条指令甚至几十条指令。这个长指令字在指令译码站 ID 一起译码,然后在执行站分别执行,图 1(d)中是执行 3 条指令。多发射结构还有其他的类型,例如把超标量与超流水线结合起来,称为超标量超流水线结构;又例如数据流计算机(参见**数据流计算机**),也是一种多发射结构。



IF —— 取指令; ID —— 指令译码;
EX —— 执行指令; WR —— 写回结果

图 1 单发射结构和多发射结构的指令执行流水线

超标量结构在发射出多条指令给执行部件以后,在执行站要同时执行这多条指令,所以要求处理机内具有多个执行部件,来完成不同类型的多条指令的同时执行。超流水线结构发出的多条指令,在

执行部件中一条接着一条地执行，但在每个子周期必须执行出结果，因此要求执行部件的工作周期较短。可以说，超标量结构是以空间换取时间，而超流水线是以时间换取空间。由于20世纪90年代初的半导体芯片上已可以集成几百万个晶体管，所以当时的大部分多发射结构都是超标量结构。后来VLSI电路的线宽已降得更细，到几十纳米，芯片的工作电压降为3.3V甚至更低，这些都有利于芯片工作频率的提高，因此20世纪90年代中期，多发射结构已采用超标量超流水线结构。超长指令字结构可开发更高的指令级并行性，但由于其指令格式较难与已有的计算机做到二进制兼容，另外，编译优化的难度也较大，所以超长指令字多发射结构未能被广泛采用。

多发射结构的目的是提高指令执行的并行度，即在维持与已有处理机二进制兼容的情况下使处理机的处理速度升级。多发射结构的出现，对于编译优化技术的要求也提高了，尤其是要解决指令级的转移相关性与数据相关性问题。这些相关性问题使指令执行并行性的提高受到约束。为了解决这一瓶颈，需要发展编译优化技术与指令级并行处理技术。

（李三立）

duo fenbianlü zaoxing

多分辨率造型（multi-resolution modeling） 在**几何造型**的基础上，将给定模型生成不同细节的表示。常用于生成多分辨率模型。

给定一个基网格 X_0，在 X_0 上施加一系列修改操作（比如简化操作）$M_1, M_2, \cdots, M_k$。这些操作之间的依赖关系将 $X_0, M_1, M_2, \cdots, M_k$ 组织成一个有向无循环图（directed acyclic graph），称为 X_0 的**多分辨率模型**。由它可以生成符合指定分辨率要求的模型。

随着数据获取设备和CAD造型系统的发展，生成的模型往往包含高度复杂的细节。为了在合理的时间内对复杂模型进行有效处理，需要对它的形状进行不同程度的近似，生成多分辨率模型。它常用于对复杂场景的实时绘制，可以有效地控制场景复杂度，加速图形绘制，提高交互性。在CAD领域，边界表示模型和特征模型也可被转化为多分辨率模型，以适应不同的处理需求。

多分辨率模型的生成算法主要有以下几种。①基于小波的方法：小波技术为多分辨率分析提供了一种简单、一致的方法来处理复杂网格在存储、传输、绘制以及编辑处理中的问题。②基于网格简化或者网格细分的方法：这种方法生成的多分辨模型，一般采用层次或序列结构组织网格简化或细分中生成的数据。以网格简化为例，层次结构将每次简化操作新生成的图元作为父结点，将被删除的图元作为子结点，从而将数据组织成一棵层次树；序列结构将所有的数据组织成序列，排在后面的记录是从前面的记录中简化而来的。③特征抑制的方法：在CAD领域，经常通过抑制模型的特征，生成多分辨率模型。

递进网格（PM）是一种多分辨率模型表示，采用树结构记录新生成的结点与被删除的结点之间的父子关系，从而支持有选择的精化和简化操作。另一种多分辨率表示方法是黏合多分辨率模型（GMM），GMM把在三维空间内有邻接关系、出现在层次树不同层上的三角形编码在一起，以便快速地访问拓扑结构上相邻的三角形以及在简化过程中相邻的三角形，为任意区域中网格的简化和精化提供了统一的操作模型。

在基于多分辨率模型的绘制算法中，如果一个物体离视点较远，或者这个物体较小，就用较低分辨率的模型进行绘制；反之，如果一个物体离视点较近，或者物体较大，就必须用较高分辨率的模型进行绘制。多分辨率模型可用于支持**实时绘制**、几何模型压缩和传输，在**虚拟现实**、交互式可视化、飞行仿真、协同设计等系统中取得了成功的应用。

参考文献

石教英，等. 虚拟现实基础及实用算法. 北京：科学出版社，2002

（潘志庚　蔺宏伟）

duohe chuliqi

多核处理器（multi-core processor） 在一个处理器的内部集合了多个运算核，能够同时并发执行多条处理器指令的处理器形式。在计算机中，一个运算核一般被认为是一个能够读取并且执行程序指令（即运算）的单元，该单元还可能读取指令的源数据并输出指令的计算结果数据。这里的多核处理器是相对于单核处理器而言的。多核处理器一般包含有两个或者两个以上的计算核心，若具有更多的计算核心则称为**众核处理器**。多核处理器和众核处理器的计算核心数目的区分没有明显的界限：一般认为数目达到数十个就可以被称为众核处理器，而在此数目之下被认为是多核处理器。在生产工艺上，通常将多个计算核心集成到单一的集成电路硅

片上，被称为片上多处理器，也有可能放到多个集成电路硅片，但是通过工艺构成单个处理器的封装。

多核处理器与众核处理器的出现是由于芯片工艺的发展表明在提高单个核心处理器的运行速度上出现了瓶颈。主要是：功耗受限制、内部互连网络延迟以及设计更加复杂。在功耗限制方面，单纯提高频率的方式会带来处理器的热量问题，无法在实际工作中有效散发所产生的热量，不能保证处理器的稳定运行。在处理器内部互连网络延迟方面也制约了单个核心处理速度的进一步提高，由于光速的有限特性，限制了处理器中计算速度的提高。在处理器内部，需要多个单元共同协作来完成指令执行的任务，因此需要将计算结果从处理器的一个部分传送到另外的部分，这就造成了线路的延迟。随着片上晶体管数目的急剧增加，这部分线路的延迟也造成了不可忽视的影响。最后，随着晶体管数量的增加，处理器的设计空间、相应的设计复杂度以及为了保证处理器正确运转的处理器验证几方面，如果仍然采用单核的设计，无疑会造成很大的困难。因此，现有的处理器生产厂商通过多核以及众核的方式来提高处理器的性能。多核处理器与众核处理器是通过核心数目的横向扩展来达到提高处理器性能的目的，这与传统的通过体系结构改进、核心频率升高等方法进行纵向扩展单个核心的性能方法不同。程序在多核处理器上的执行是通过将程序进行任务划分，让不同的指令流在不同的核心并行执行的方式来达到提高处理器性能的目的。

多核处理器与众核处理器的分类

根据核心类型是否相同，多核处理器与众核处理器一般被划分为同构的多核处理器以及异构的多核处理器。

同构多核处理器中的每一个核都是相同的，没有主次之分，功能也一致。有些同构多处理器的每一个核心都是完整的处理单元，能够担负整个系统中的各种处理任务，包括控制以及计算，这包括从传统的处理器升级的多核处理器，能够直接用于搭建计算机。另外一些同构处理器用于特殊的应用，本身缺乏一定的全系统控制能力，必须依附于其他的中央处理器才能够用于搭建计算机，这包括传统显卡发展的通用**图形处理器**。

异构的多核处理器各个核之间有功能的划分，处理器内部核心存在差异，互不相同。异构多核处理器中有些内核是原有的中央处理单元发展过来的，具有完整的系统控制与处理能力，能够用于独立搭建计算机。另外一些内核不具有完整的控制能力，是具有特殊功能的计算加速器，能够加快某些计算的速度。异构处理器的一个典型例子就是 IBM 开发的 Cell 处理器，在 Cell 处理器中具有一个 PowerPC 的处理器内核，这是通用的处理器内核，具有完整的中央处理单元功能；在 Cell 处理器还有 8 个浮点处理单元，这 8 个浮点处理单元具有很高的浮点处理能力，对于某些应用具有很好的加速能力，例如游戏以及科学计算，因为这些应用有很多的浮点运算。这样 Cell 的 9 个核被分成一个通用的处理器以及 8 个浮点加速单元，构成了异构多核处理器。除了固有的核之外，异构处理器有时还会包括一些可以编程的核心，例如 FPGA 的核心，可以通过编程的方式构造面向应用的特定处理器，以提高对应应用的执行能力。

采用多核处理器与众核处理器的影响

通过多核以及众核的方式，软件能够有机会更快的运行。由于多个核心处于一个芯片的内部，使得处理器之间的通信速度大大加快，原有的用于多处理器之间进行互连的技术被用于芯片的内部。运行于芯片内部的互连网络可以大大简化，并且能够提高性能。信号在片上多个内核之间的传输要远远快于通过主板进行通信的多个处理器，这样会加速缓存一致性的效率。由于在单个芯片上放置了多个处理器的内核，使得并发多任务成为可能，应用程序可以运行于不同的处理器内核之上，相互之间可以降低性能上的影响，真正实现并发执行。

在产品设计上，使用多核的方式可以让厂商能够减少印刷电路板的面积，原先用于连接多个处理器的印刷电路板可以进行简化，并且提高了性能。另外，通过多核而不是多个处理器的方式，可以使用简单设计的多个内核，而不是复杂设计的单个核心，在提高性能的同时，还可以降低能源的消耗，大幅提高系统总体的能耗比。在芯片设计上，多核设计也可以简化设计的成本，提高设计的成功率，因为可以使用现有的已经通过验证的内核。

但是，为了获得上述的多核处理器与众核处理器带来的好处，软件（包括操作系统软件以及应用层软件）必须做出改变，才能够充分利用多核处理器与众核处理器的计算能力。在多核处理器与众核处理器条件下，软件必须使用多线程等技术来达到并发的并行性，这要求并发程序设计人员必须要考虑到自己的代码会同时运行在多个处理核心之上，这比设计单线程的程序要困难许多。并发程序设计

的内容包括:①将数据以合适的方式划分到多个处理器内核中;②在程序运行的过程中,要进行多个核之间的通信,以推动程序的执行;③由于使用了共享内存的方式,对共享资源进行保护的核间同步则是避免不了的手段。另外,多核上运行的多线程的程序通常具有不确定性,即同一个程序的两次执行行为是不一样的,这就给应用程序的调试带来了很大的困难。

参考文献

1. Hennessy J L, Patterson D A. Computer architecture: a quantitative approach, 5th ed. Morgan Kaufmann, 2011

2. 多核系列教材编写组. 多核程序设计. 北京:清华大学出版社,2007

(陈康)

duo jiqiren xitong

多机器人系统(multi-robot system) 若干个机器人通过合作与协调完成问题求解任务的系统。它不是物理意义上的单个机器人的简单相加,也不是单个机器人作用效果的求和。多机器人系统作为一种人工系统,实际上是对自然界和人类社会中群体系统的一种模拟。基本思想是将多机器人系统看作是一个群体或一个社会,从组织和系统的角度研究多机器人之间的合作与协调机制,充分发挥多机器人系统各种内在的优势。

多机器人系统虽是从单个机器人系统扩展而来,但区别于单个机器人系统,其特点可以概括如下:

(1) 空间分布 多个机器人可以在工作空间的不同区域同时工作。

(2) 功能分布 功能或者任务不同的多机器人可以协调工作。

(3) 时间分布 多机器人可以执行时间分布的任务。

(4) 信息分布 具备相同或者不同知识的多机器人,通过通信,可以进行知识的交换和学习。

(5) 资源分布 多机器人系统中各个机器人可以具有不同的传感器和执行器。

多机器人系统的上述特点,使得它在应用中能够达到以下目的:

(1) 利用其空间分布特性,通过多机器人并行作业可以提高完成任务的工作效率。

(2) 利用系统内各种资源(知识、信息等)的共享能够弥补个体能力的不足,提高完成任务的能力。

(3) 利用系统内资源的冗余性以及各个机器人功能的互补性,可以提高完成任务的可能性,增强系统的容错性、鲁棒性和灵活性。

(4) 利用多机器人系统功能分布、资源分布的特点,可以降低单个机器人的设计成本和难度。

虽然多机器人系统在功能、结构及应对复杂环境和各种任务等方面具有许多优点,但它也存在一些问题:

(1) 分布式的结构使得系统全局优化变得复杂,甚至无法获得最优解。

(2) 随着机器人数量的增加,多机器人之间进行组织和相互协调与合作的困难也以指数级增长。

(3) 多机器人系统与单个机器人相比,虽然可以提供更多的解决方案,可以针对不同的具体情况,优化选择方案,但是同时也增加了机器人对其可用功能、信息和资源进行搜索、交换和分配的时间。由此导致的通信问题降低了系统响应外界环境变化的能力。

多机器人系统的研究涉及了**人工智能**、计算机、自动控制、机械电子等多门学科,其主要研究内容有:

群体体系结构 多机器人系统依靠几个机器人的简单组合不能发挥其优势,只有通过某些形式的合作才能实现对复杂任务的处理。而机器人要实现相互之间的良好合作和协调就必须确定它们逻辑上和物理上的信息和控制关系,以及问题求解能力如何分布等问题。针对这些问题进行的群体体系结构研究可以将多机器人系统的结构和控制有机地结合起来,保证信息流和控制流的畅通。合理的结构可以使多机器人之间进行高效的合作。多机器人的群体结构可以分为集中式和分散式两种。分散式又可进一步划分为分层式和分布式结构。

感知 机器人的感知包括感觉和理解两个方面的问题。一般地,机器人配备各类不同功能的传感器使机器人“感受”机器人本体和外部环境的变化信息。通过对这些信息进行有效的融合、处理及解释,机器人能够“理解”它们所包含的意义,并将它们与机器人的决策和控制紧密地结合起来。多机器人信息感知是实现机器人之间合作与协调的基础。

通信 是机器人之间进行交互和组织的基础。通过通信,多机器人系统中各机器人能够了解到其他机器人的意图、目标以及当前环境状态等信息,进而进行有效的磋商和协作,共同完成任务。机器人之间的通信分为隐式和显式通信两类。

学习 由于单个机器人工作能力的限制，其所感知的环境是局部的、片面的，个体机器人为了学到较全面的知识以适应外部环境的各种变化需要花费大量的时间。通过多机器人系统进行学习，各机器人则可以“交流”彼此学到的知识，分享各自的经验。

协作机制 用于解决多机器人之间的合作与协调问题。当给定多机器人系统某项任务时，首先面临的问题是如何组织多个机器人去完成任务，如何将总体任务分配给各个成员机器人，也即机器人之间怎样进行有效地合作。当以某种机制确定了各自任务与关系后，问题变为如何保持机器人间的运动协调一致，即多机器人协调。多机器人合作和协调是多机器人系统研究中的两个不同而又有联系的概念。前者研究的重点是高层的组织与运行机制问题，侧重实现系统快速组织与重构的柔性控制机制；后者则是研究机器人之间合作关系确定后具体的运动控制问题。

多机器人系统的研究随着20世纪80年代分布式人工智能和复杂系统的研究工作逐渐开展而活跃起来，并被作为实验平台进行相关的理论论证和仿真。多机器人系统的研究将多智能体、社会学和管理学等其他领域的理论和方法引入到机器人学的研究中，丰富了机器人学研究的内容。在多机器人系统研究中，以机器人足球赛最具代表性。在机器人足球赛中（如图1），同队的各个机器人之间是合作的、互助的，不同球队的机器人之间的关系则是对抗的、竞争的。

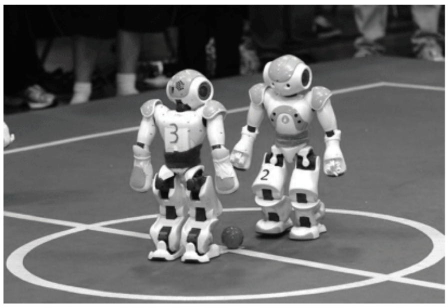


图1 机器人足球赛

参考文献

谭民，等. 多机器人系统. 北京：清华大学出版社，2005

（杨唐文）

duolei daishu

多类代数（many-sorted algebra） 在通常的代数中引进类型符号而形成的代数理论。最初由Bénabou于1968年以范畴论的形式提出。现在，多类代数已成为**形式语义学**的主要理论工具，它在软件规约、程序正确性证明等领域有重要应用。

若S为类型符号集合，Σ为运算符号集合，并且Σ中每个运算符的类型形如$s_1,\cdots,s_n\to s_0(n\geqslant 0)$，$s_i\in S$，则称$\langle S,\Sigma\rangle$为基调。设$A=\{A_s\mid s\in S\}$为集族，$F$为函数族，如果对$\Sigma$中任意$\sigma:s_1,\cdots,s_n\to s(n\geqslant 0)$，都有$F$中唯一的函数$\sigma_A:A_{s_1}\times\cdots\times A_{s_n}\to A_s$，则称$\langle A,F\rangle$为一个$\Sigma$-代数。所有$\Sigma$-代数的全体构成多类代数族。

给定基调$\langle S,\Sigma\rangle$及Σ-代数$\langle A,F\rangle$和$\langle A',f'\rangle$。从$\langle A,F\rangle$到$\langle A',F'\rangle$的Σ-同态是满足以下条件的函数族$\{h_s:A_s\to A'_s\mid s\in S\}$：对$\Sigma$中任意运算符$\sigma:s_1,\cdots,s_n\to s(n\geqslant 0)$，有$h_s(\sigma_A(x_1,\cdots,x_n))=\sigma_{A'}(h_{s_1}(x_1),\cdots,h_{s_n}(x_n))$，$x_1\in A_{s_1},\cdots,x_n\in A_{s_n}$。由全体$\Sigma$-代数和$\Sigma$-同态构成的范畴的初始对象和终止对象分别称为初始代数和终止代数。

Σ-代数$\langle A,F\rangle$上的同余关系$\equiv$为满足下述条件的关系族$\{\equiv_s\mid s\in S\}$：①对任意$s\in S$，$\equiv_s$是A_s上的等价关系；②若$a_i\equiv_{si}a'_i$，则对Σ中任意$\sigma:s_1,\cdots,s_n\to s(n\geqslant 0)$，皆有$\sigma_A(a_1,\cdots,a_n)\equiv_s\sigma_A(a'_1,\cdots,a'_n)$。基于同余关系$\equiv$可得到$\langle A,F\rangle$的商代数$A/\equiv=\langle\{A_s/\equiv_s\mid s\in S\},F/\equiv\rangle$，$F/\equiv$中的函数定义如下：对$\Sigma$中任意$\sigma:s_1,\cdots,s_n\to s(n\geqslant 0)$及$a_i\in A_{s_i}$，有$\sigma_{A/\equiv}([a_1]_{\equiv s_1},\cdots,[a_n]_{\equiv_{s_n}})=[\sigma_A(a_1,\cdots,a_n)]_{\equiv_s}$。

给定基调$\langle S,\Sigma\rangle$及变元集族$X=\{X_s\mid s\in S\}$。基于X的自由生成Σ-代数$T_\Sigma(X)=\{T_{\Sigma,s}(X)\mid s\in S\}$构造如下：①$X_s\subseteq T_{\Sigma,s}(X)$且对$\Sigma$中任意$\sigma:s_1,\cdots,s_n\to s(n\geqslant 0)$，若$t_i\in T_{\Sigma,s_i}(X)$，则$\sigma(t_1,\cdots,t_n)\in T_{\Sigma,s}(X)$；②对$\Sigma$中任意$\sigma:s_1,\cdots,s_n\to s(n\geqslant 0)$及$t_i\in T_{\Sigma,s_i}(X)$，$\sigma_{T_\Sigma}(X)(t_1,\cdots,t_n)=\sigma(t_1,\cdots,t_n)$。特别地，若$X=\varnothing$，则记$T_\Sigma(X)$为$T_\Sigma$，称为$\Sigma$-项代数或自由字代数。对任意函数$h:X\to A$，存在唯一的$\Sigma$-同态$\bar{h}:T_\Sigma(X)\to A$扩充$h$，即$\bar{h}(x)=h(x)(x\in X)$。

抽象数据类型的语法形式为$\langle S,\Sigma,E\rangle$，其中$E$为$T_\Sigma(X)$中的项所形成的等式或条件等式公理。满足$E$的$\Sigma$-代数称为$\langle S,\Sigma,E\rangle$的语义模型。所有模型所构成的范畴的初始对象和终止对象分别称为$\langle S,\Sigma,E\rangle$的初始语义和终止语义。前述$T_\Sigma$的重要意义在于：经由$E$导出的某些同余关系的划分得到的$T_\Sigma$的商代数可以构成$\langle S,\Sigma,E\rangle$的初始语义和终止语义。在初始语义中成立的许多性质可推广至抽象数据类型的其他语义模型。

为了以结构化方式描述抽象数据类型，需要引进层次协调性和充分完全性。对于抽象数据类型 $ADT_i = \langle S_i, \Sigma_i, E_i \rangle (i=1,2)$，如果 $S_1 \subseteq S_2, \Sigma_1 \subseteq \Sigma_2$，$E_1 \subseteq E_2$，则称 ADT_2 是 ADT_1 的扩充。ADT_2 相对于 ADT_1 的层次协调性是指：对 ADT_1 的任意项 t, t'，$t = t'$ 在 ADT_1 中可证当且仅当 $t = t'$ 在 ADT_2 中可证。充分完全性是指：对 ADT_2 中任意项 t_2，如果 t_2 的类型在 S_1 中，那么必存在 ADT_1 中的项 t_1，使得 $t_1 = t_2$ 在 ADT_2 中可证。为了保证语义上的合理性，在对抽象数据类型进行扩充或组合时，必须考虑层次协调性和充分完全性。

为了处理计算机科学和软件工程中的复杂问题，人们对多类代数进行了各种扩充，以便处理部分函数、高阶函数、异常机制、子类型机制、非确定性和并发性。例如，序类代数就是在多类代数的类型符号集合上引进偏序关系得到的代数理论。

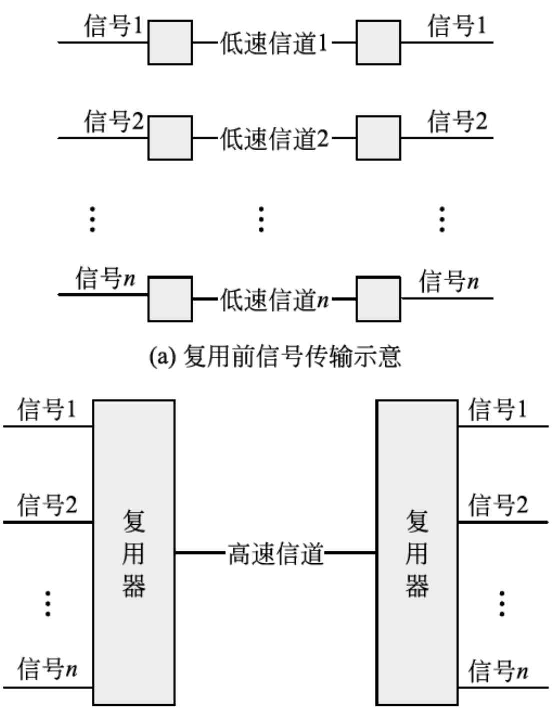
参考文献

陆汝钤. 计算系统的形式语义. 北京：清华大学出版社，2017

（谭庆平）

duolu fuyong jishu

多路复用技术（multiplexing technology） 指在一条高速信道上同时传输多路信号的技术。多路复用技术原理如图 1 所示，其中发送端的多路低速信号首先接入多路复用器，汇集后组成一路高速信号，通过一条高速信道进行传输；接收端在收到高速信号后，送入复用器（也称解复用器），还原成多路低速信号，完成复用。即通过一条高速信道代替多条低速信道。



(a) 复用前信号传输示意

(b) 多路复用技术示意

图 1　多路复用技术的示意图

从图 1 可以看到，采用了多路复用技术，可以节约传输信道和通信介质；同时，多路复用系统对终端用户是透明的，复用和解复用的工作由复用器完成，终端用户不需做额外的工作。

多路复用技术主要有时分多路复用（TDM）、频分多路复用（FDM）、波分多路复用（WDM）、码分多路复用（CDM）和空分多路复用（SDM）等。

参考文献

Tanenbaum A S. 计算机网络. 北京：清华大学出版社，2004

（袁华　张凌）

duomeiti caozuo xitong

多媒体操作系统（multimedia operating system） 支持多媒体设备及其多媒体数据处理的操作系统。

由于多媒体数据具有数据量大、类型多样、高数据率、实时传输、时间和逻辑依赖性强等特点，与传统通用操作系统相比，多媒体操作系统在功能和性能上有其自身特点，主要有：

必须提供可定制的多媒体设备驱动，定义可扩展的媒体控制接口，实现不同媒体设备的控制服务，为用户提供丰富的多媒体交互环境。

进程管理中必须提供 连续媒体数据的实时处理及其时序要求的功能；进程调度常采用可抢占的实时调度，支持硬实时与软实时进程，通信与同步机制常采用时间驱动的管理方式。

提供多媒体数据文件管理能力，具体包括具有满足实时性和大数据量内容存放需求的文件系统布局和文件格式，支持高速缓存和高效磁盘调度等。

具有灵活的多媒体应用编程接口，提供捕捉、存储、管理、同步和表现连续媒体数据的能力，支持高层多媒体信息的采集、编辑、播放和传输等，为应用开发者提供高级服务。

多媒体操作系统的实现方式可以分为传统操作系统扩展和重新设计微内核结构两种方式。

参考文献

1. Tanenbaum A S. 现代操作系统. 陈向群，马洪兵，等译. 北京：机械工业出版社，2005

2. Steinmetz R. Analyzing the multimedia operat-

ing system. IEEE MultiMedia, 1995

（周兴社　张羽）

duomeiti jishu

多媒体技术(multimedia technology)　为了把文本、图形、图像、声音、动画、视频等多种形式不同而逻辑上相关的内容向用户进行展现而对它们进行综合处理的一种技术。“多媒体”是形容词,一般应与名词联用。例如,具有上述特性的计算机就是多媒体计算机,具有上述功能的通信系统就是多媒体通信系统,能够有效地储存、管理、检索多媒体信息的数据库系统就是多媒体数据库系统。

多媒体技术强调的是交互式综合处理多种信息媒体的技术。从本质上来说,它具有 3 种重要的特性:

第一是信息媒体的多样性,它不仅能处理传统的数值、文字、静止图像(静态媒体),更重要的是能有效地处理言语、音乐、动画和视频等随时间变化其内容的动态媒体;

第二是多种媒体的集成性(即综合性),它使用多种不同的信息媒体(特别是它能把静态媒体和动态媒体进行组合)综合地表现某个内容,取得更好的表现效果;

第三是内容展现过程中的交互性,在使用多媒体信息展现内容的过程中,用户可以对系统进行有效的、细粒度的控制,使人们获取和使用信息变被动为主动。

由于多媒体信息的数据量极大,特别是视频信息和音频信息,数据类型繁多,处理复杂,且往往有实时或限时处理的要求,不同的媒体相互间还必须提供同步机制等,因此多媒体技术比常规的信息处理技术要复杂得多。

多媒体技术是围绕着多媒体作品(也称为“多媒体文档”)展开的。一部数字影片、一段动画、一个计算机辅助教学(CAI)课件、一段远程教学的直播场景等都是多媒体作品。从用户角度看,多媒体信息处理的过程可分成图 1 所示的若干不同阶段,整个过程称为“媒体食物链”。

(1) 媒体准备　媒体准备由多媒体获取设备及相关软件完成,也可以人工创建。例如文字输入、图形绘制、**图像获取**、声音录制、视频拍摄等,这是把媒体引入计算机数字世界的关键。不同类型的媒体具有不同的性质,所使用的硬件设备及其处理软件各不相同,但目的都是获得或是人工创建数字媒体。

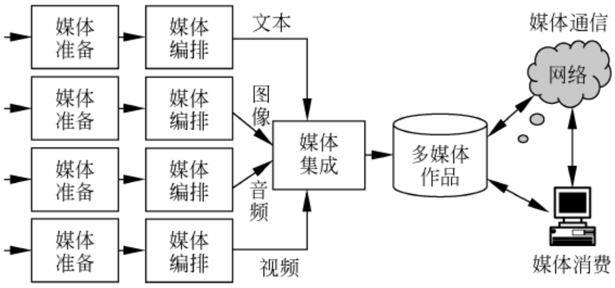


图 1　多媒体信息处理过程

(2) 媒体编排　对各个媒体进行编辑和排版处理,如将不同的组成对象合成一体,修改其组成对象的内容(如字符、语句、配音、视频画面)及属性(如文字的字体、语句的速度和图像的颜色等)。这个阶段需要使用各种数字媒体所特有的编辑软件来完成,如文字处理软件、绘图软件、图像处理软件、声音编辑器、音序器软件、非线性视频编辑器等。

在媒体获取与编排的过程中,一个关键的问题是各种数字媒体如何表示(即媒体的编码)。数字媒体编码的目的之一是为了压缩数据、提高编码效率,以减轻传输和储存的压力;另一个目的是为了合理地组织数据,以满足各种应用的要求,包括在各种不同系统中的交换和互操作。为此,国际标准化组织(ISO)、国际电工委员会(IEC)和国际电信联盟(ITU)等制定了一系列技术标准,如静止图像编码标准 JPEG,JPEG—2000;运动图像及其伴音的编码标准 MPEG-1、MPEG-2 和 MPEG-4 等。

(3) 媒体集成　为了把数字媒体对象相互关联构成一部多媒体作品,必须按作品脚本的规定正确处理各媒体对象相互间的时空关系。媒体集成涉及的技术问题很多,例如多媒体作品的数据模型、规范与标准,媒体的空间布局与时间同步,媒体的说明、标识与描述,媒体的存储组织与管理,目录与索引,以及多媒体作品知识产权的管理与保护等。用于媒体集成的软件是多媒体写作工具软件。

(4) 媒体通信　媒体通信指在网络上传递、分发和交换多媒体作品,为最终用户提供远程的多媒体信息服务(例如视频会议、协同工作、电子邮件、信息检索等)。它既是技术手段,也是应用。为了进行多媒体数据通信,网络必须提供足够的带宽,满足规定的传输延迟时间和端—端的抖动幅度等要求,提供不同的媒体数据流之间的同步机制,并能支持多播和广播方式的网络通信等,这就给通信技术和**计算机网络**带来了严峻的挑战。

（5）媒体消费　媒体消费即多媒体作品的应用，通常有两种方式：单机应用（本地应用）方式和网络应用（分布式应用）方式。单机应用方式如 CAI 课件、电子游戏、视频光盘（DVD）等。网络多媒体应用是多媒体应用的主流，是发展方向，它技术难度大，但也是影响最大、最有效益的应用方式。网络多媒体应用又可以分成两大类：①面向人与人通信的网络多媒体应用（p-to-p 方式），此类应用过程至少有 2 个人介入，其目标是改善人们远程通信的效果。例如 IP 电话、实时远程教学、视频会议、网上电子游戏、多媒体电子邮件、多媒体文档交换、计算机支持的协同工作（CSCW）等。②面向人与系统通信的网络多媒体应用（p-to-s 方式），此类应用的目标是改善或者提供创新的用户与信息源之间的通信方式，例如网上购物、视频点播、网上电视（Web-TV）、网上广播（Web-radio）等。

多媒体技术是信息处理技术发展的一种必然，它的发展推动了许多相关产业的改造和调整，使计算机、通信、广播、出版、消费类电子产品等在技术上逐步走向融合。随着网络多媒体应用的飞速发展，人们的工作和生活方式将发生显著改变，多媒体技术也会得到进一步的发展。

参考文献

1. 林福宗. 多媒体技术基础. 3 版. 北京：清华大学出版社，2010

2. 胡晓峰，等. 多媒体技术教程. 3 版. 北京：人民邮电出版社，2009

（张福炎）

duomeiti kuozhan bujian

多媒体扩展部件（multimedia extension unit，MMXU）　处理器内部用于执行多媒体等应用中数据并行计算的功能单元。与浮点运算部件和定点运算部件不同，多媒体扩展部件能一次并行执行多个短字长数据单元的运算操作。多媒体扩展部件对应着一组多媒体扩展指令，其操作数为固定宽度（一般为 64 位到 256 位）的长字。通过对长字内数据单元字长的配置（如字节、半字、字、双字等），编程人员可以实现长字内多个数据单元的并行运算。随着多媒体应用的日益普及和处理器能力的逐步提升，几乎所有的通用处理器都配备了多媒体扩展部件。

多媒体扩展部件最早由惠普公司于 1994 年在 PA-7100LC 上实现，其多媒体加速扩展（Multimedia Acceleration eXtensions，MAX）可支持 32 位机上的 16 位整数加减和移位运算。随后的 MAX-2 把 MAX 扩展到 64 位机上。

1996 年 Intel 将多媒体扩展（MultiMedia eXtension，MMX）引入到 Pentium 处理器中。MMX 指令的操作数为 64 位，可以被配置成 1 个 64 位整数、2 个 32 位整数、4 个 16 位整数或 8 个 8 位整数。1999 年 Intel 推出 SSE，对 MMX 进行了加强。SSE 指令的操作数为 128 位，支持单精度浮点和定点的混合运算。此后 Intel 又相继推出了 SSE 2、SSE 3 和 SSE 4，实现了对双精度浮点运算的支持。

其他通用处理器也先后引入了多媒体扩展部件，如 PowerPC 处理器的 AltiVec/VMX 部件、AMD 处理器的 3DNow 部件，等等。

通常，图像、声音等信息的基本数据只需要 8 位或 16 位即可表示，用 32 位或 64 位的处理器字长来处理这样的短数据是很不经济的。如果把一个 64 位加法器的进位链划分为 8 个 8 位的段或 4 个 16 位的段，就能以少量硬件和功耗的增加，同时完成 8 个 8 位或 4 个 16 位的加法。这是设计多媒体扩展部件的基本出发点。从原理上看，多媒体扩展部件是一种以单指令流多数据流（SIMD）方式开发数据级并行性的运算部件。与普通的运算部件一样，多媒体扩展部件主要包括寄存器和运算单元。在执行指令时，多媒体扩展部件从寄存器中取操作数，进行运算后写回寄存器或主存中。早期的多媒体扩展部件（如 MMX）与 CPU 的浮点协处理器共用寄存器，因此会影响浮点运算。现代的多媒体扩展部件都拥有自己独立的运算单元和寄存器等，可与处理器内的其他运算部件同时工作。

随着微电子工艺的进步以及芯片集成度的提升，多媒体扩展部件的寄存器数目和操作数长度不断增加，其处理能力和并行度也不断提高。编译技术的发展，特别是程序的向量优化技术的发展，为多媒体应用的向量优化提供了强大的技术支持，同时为多媒体扩展部件的发展提出新的要求。多媒体扩展部件逐渐脱离中央处理器（CPU）结构，向着功能更独立，向量化能力更大的加速器方向发展（参见**向量处理部件**）。

另一方面，随着移动多媒体应用的发展，多媒体扩展部件已经开始应用到嵌入式处理器，如 ARM 公司已经将新型的多媒体扩展部件 NEON 加入手机芯片 Cortex-A8/A9 中。由于嵌入式领域对极低能耗的需求，多媒体扩展部件的发展将在计算能力和低功耗上进行权衡。

参考文献

1. Diefendorff K, Dubey P K, Hochsprung R, Scales H. AltiVec Extension to PowerPC Accelerates Media Processing. IEEE Micro, 2000, 20(2): 85-95

2. Brey B B. Intel 微处理器. 8 版. 金惠华,等译. 北京:机械工业出版社,2010

(于俊清 魏海涛)

duomeiti shujuku

多媒体数据库(multimedia database) 以多媒体数据存储和多媒体信息处理为中心的数据库。多媒体数据包括数字图像、音频、视频、动画、图片等广泛的数据类型。近年来,多媒体数据的产生、获取、存储和处理已成为计算机应用领域不可忽视的重要需求。各种多媒体应用期望借助数据库系统管理和维护海量的多媒体数据,为用户提供便捷的**查询**或**检索**功能。

多媒体数据库系统必须提供统一的存储、处理、检索、传输和展现多样的多媒体数据的平台,并为这些不同类别的多媒体数据提供传统**数据库管理系统**的服务和面向多媒体数据的管理功能。多媒体数据的引入,给数据库技术带来了众多的挑战,包括:海量数据、时基特征、语义丰富性、表现复杂性、解释主观性、应用多样性等。因此,目前多媒体数据库仍处于发展阶段,仅在某些特定应用领域得到了良好运用,尚缺乏通用的多媒体数据库系统。

多媒体数据库系统的设计需要满足以下需求:①支持多种类型的输入、输出和存储设备;②多种类型的数据压缩和存储格式的处理能力;③不同类型多媒体数据模型的集成;④友好、高效的用户查询和检索接口;⑤多种类型的索引,支持多媒体数据的非精确查询;⑥符合用户感知体验的媒体数据相似度评估模型;⑦满足实时展现要求的数据传输与多类别数据的同步能力。

多媒体数据库设计需解决以下问题:

(1)数据库的组织和存储 多媒体数据的数据量大,不同媒体的数据规模差异也大,需要数据库选择合适的物理结构和逻辑结构,以保证磁盘的充分利用和应用的快速存取。

(2)媒体种类的增加 不同媒体类型对应不同的数据处理方法,要求多媒体数据库管理系统能够不断扩充新的媒体类型及相应的操作方法,并且新媒体类型的增加对用户应该是透明的。

(3)多媒体数据库的查询 与传统数据库精确查询的概念不同,多媒体数据库中广泛存在着非精确匹配和相似性查询。由于媒体语义解释的主观性,使其在语义级别上查询、检索和融合各种媒体类型的数据变得异常困难。

(4)用户接口的支持 多媒体数据的丰富性和表达的多样性要求多媒体数据库提供与传统数据库系统不同的用户接口。多媒体数据库的用户接口不仅仅是接受用户的描述,而且还要辅助用户形成反映其想法的查询或检索条件,找到候选的内容,并在接口上表现出来。

(5)长事务处理能力、服务质量要求和多版本管理 海量数据、时基特征导致多媒体数据库必须增加处理长事务的能力。广泛的应用对多媒体数据库的传输、表达和存储方式的质量要求也将不同。多媒体数据库系统提供的资源要根据系统运行的情况进行控制,并针对不同的应用需求和资源条件提供不同版本的多媒体对象。

参考文献

1. 百度百科. 多媒体数据库. http://baike.baidu.com/view/297200.htm

2. IT portal. Multimedia dabase. http://www.peterindia.net/Multimedia Database.html

(汪卫 施伯乐)

duomeiti wendang

多媒体文档(multimedia document) 包含多媒体数据的文档。它特指在媒体数量、媒体类型和媒体集成度三个测度上都有充分体现的文档。多媒体文档要能够表示不同媒体数据的构造及其属性特征,能够指出不同媒体数据之间的相互关系,特别是反映高层应用语义的时空特性关系。

多媒体文档要体现:

(1)数据多样性 多媒体文档必须支持多种数据类型,包括对字符文本、向量图形、位图图像、数字音频、动画、数字视频等与应用无关的通用数据类型的准确定义和对与应用有关的数据类型提供建模工具及在演示时将它们映射到基本数据类型的说明。

(2)同步特性 多媒体系统中存在着一个重要的问题是表现(presentation),这是多媒体系统所特有的,它不同于一般系统的显示,而是多种媒体数据的合成再现、加工再现、有交互性参与的创作再现。这其中同步问题被认为是多媒体系统的一个重要特征,分布式多媒体系统中由于数据分布、网络资源有限造成的不确定性使同步问题更为突出。

(3) 交互特性　多媒体的一个关键特性是交互性,它将向用户提供更加有效的控制和使用信息的手段和方法,同时也为应用开辟了更加广阔的领域。交互可以做到自由地控制和干预信息的处理,增加对信息的注意力和理解。当引入交互后,活动本身作为一种媒体便介入了信息转变为知识的过程。媒体数据的简单检索与显示是多媒体的初级交互应用;通过交互特性使用户介入到信息的活动过程中,才达到交互应用的中级水平;当用户完全进入到一个与信息环境一体化的虚拟信息空间自由遨游时,才进入交互应用的高级阶段。

(4) 组织结构　文档包含文档结构和文档内容两方面的含义,文档结构描述文档中单个元素的连接关系或文档内容的组织方式。多媒体文档的组织结构要能反映其内容。**超文本**在高层上对节点和链的区分很适合表现文档结构的高层语义,直接反映了人类联想式思维的模式,是多媒体文档最为有效的组织形式。(徐光佑)

duomeiti wendang guifan

多媒体文档规范(specifications of multimedia document)　一些国际组织和企业联盟为便于**多媒体文档**的数据交换而制定的开放性的文档规范。常见的有 HTML, MHEG, VRML, SMIL, XML 及 HyTime 等。

HTML(超文本置标语言)是标准通用**置标语言**(ML)的一种独立于平台的格式定义,以置标(说明)来建立**超文本**、超媒体文档。HTML 于 20 世纪 80 年代末 90 年代初研制成功,目前是描述万维网 Web 页面的置标语言。HTML 文档是一种纯文本文件,可以用任何文本编辑器阅读和编辑。Web 页面中包含一系列 HTML 标记,以指示浏览器如何显示 Web 页面。当用户用浏览器与某个 Web 服务器建立连接时,Web 服务器通过网络把相应的 HTML 文件发送到用户浏览器。超文本置标语言 HTML 借助统一资源定位地址(URL)可以描述 Internet 中跨越节点的超链接,简单而实用地实现了以整个 Internet 空间为操作背景的超文本、超媒体的数据存取,且具有易于在不同表现系统上移植而保持文献的逻辑完整性的特点。

由于 HTML 主要描述的是页面的布局和外观,缺乏对内容语义的描述,另外,它的置标的集合是固定的,用户不能增加自己定义的置标。因此万维网联盟(W3C)于 1998 年初推出了被称为"第二代 Web 语言"的可扩展置标语言(XML)以解决 HTML 的固有问题,已逐渐被广泛接受。在 1998 年公布了 XML 1.0 的版本 1,在 2000 年对版本 1 中的错误做了更正,公布了 XML 1.0 的版本 2。XML 试图成为 Internet 上数据的通用格式,为了支持对互联网资源的引用和使用不同的字符集和语言,XML 还采用一组由互联网以外组织规定的子语言集合。

MHEG ISO 是多媒体与超媒体信息编码专家组的简称,也用以代表它制定的标准。1997 年公布的 MHEG-5 被批准为用于交互式多媒体交换的国际标准。目前已在数字电视和数字广播的交互服务中被广泛采用。该标准定义了结构化信息独立于系统的编码,用于存储、交换和执行多媒体演示。MHEG 遵从 OSI 的思想,基于 OSI 表示层的体系结构,使用 ASN. 1 描述数据元素和数据结构,支持多媒体同步和用户交互,希望以此作为超媒体文档在不同系统间进行实时交换的工业标准,因而特别注重交互性和多媒体同步、实时表现、实时交换、最终形式表现等几方面。

MHEG 的基础就是被称为消息处理(MH)对象的信息的独立编码表示和信息的基本单元,以便在不同应用中处理和交换所用的这些对象。这些对象可以是简单的媒体对象,或者是带有内同步和链接的不同媒体的不同分量的复合对象。MHEG 在标准的设计中采用了面向对象的方法,但它对于标准的实施却并非必需。

MHEG 中提供了一种虚坐标系统,包括一个无限长的时间轴和长度有限的 X,Y,Z 3 个空间轴,用于规定内容对象在空间和时间上的定位和相互关系。在 MHEG 中,链接表示对象之间的关系,或者说是一个条件,当源对象的状态满足链接中定义的条件时,就让目的对象执行动作。所有的对象都靠这种方式联系起来,构成网状结构。

VRML(虚拟现实建模语言)是一种用于在 Web 上构作三维多媒体和共享虚拟世界的开放式语言。VRML 的基本原理与 HTML 的基本原理一样简单,都是用一系列标记告诉浏览器如何显示一个文档,它们都是描述 Web 页面的描述语言。它与 HTML 不同的是,以 HTML 为核心的 Web 浏览器浏览的是二维世界,而以 VRML 为核心的 Web 浏览器浏览的是三维世界。在 VRML 中,以节点作为基本单位,将不同的节点以层次关系组织在一起,构成 VRML 中的场景图,使互联网用户犹如身处真实世界,在三维环境中随意探询 Internet 上丰富的信息资源。

SMIL(同步多媒体集成语言)由万维网联盟(W3C)提出的一种语言规范,提供了一种统一的语言格式用以描述多媒体数据的集成和交互。SMIL也是一种置标语言,它定义了一系列遵循可扩展置标语言(XML)规范的元素和属性,以生成包含多种媒体对象的复合多媒体文档,规定它们在时间和空间上的布局关系及运行环境等,SMIL还提供了简单的基于事件的激活方式,以实现媒体对象的交互操作。

HyTime(基于时间的超媒体结构化语言)是ISO为指定超媒体和多媒体文件的逻辑结构而定义的标准。HyTime主要研究多媒体同步的表示,超媒体在文档内或文档间的链接。HyTime适用于综合的开放型多媒体和超媒体信息系统以及在开放环境下的文档交换和操作管理。HyTime并不规定信息内容的格式、编码或文件类型,而是提供一种框架结构,用置标语言来描述多媒体文档的超级链接方式,以明确多媒体信息内容在时间上和空间上的相互关系,从而实现同步。HyTime与MHEG在很多方面是一致的,但它们的使用方法和应用环境却不同。涉及文档的处理和交换可用HyTime,涉及对象的处理与交换则用MHEG。同MHEG一样,HyTime的文档实现也很复杂。

(徐光祐)

duomeiti zhuzuo gongju

多媒体著作工具(multimedia authoring tool, authorware) 多媒体文档和多媒体应用程序的一种软件开发工具。它能够统一地编辑、管理多媒体数据,一般不需要高级语言编程就可以把这些数据连接成完整的多媒体文档或应用程序。

著作工具试图将多媒体应用开发过程中的编程工作简化,使得没有编程经验的用户(著作者)也可以利用著作工具制作能展现图象、文本、动画、音频、视频等多媒体数据并具有一定交互能力的多媒体应用程序。著作工具的主要特点首先在于它简洁易用,通常在几天,甚至几个小时培训之后,用户就可以开始进行创作。多数著作工具都提供图形界面,这些图形界面为用户设想出程序的隐含流向,并自动处理许多具体的编程指令。此外,著作工具通常都提供样板应用程序,这些样板只需进行适当改动即可快速拼装为一个应用程序。

多媒体著作工具的不足之处一般有两个:一是所开发的多媒体应用效率较低,通常要比用程序设计语言开发的多媒体应用运行速度慢;二是开发的多媒体应用的灵活性受到限制,有些特殊的或者复杂的功能著作工具并不提供。为了克服这些缺点,大多数多媒体著作工具都能支持一种或几种脚本语言,例如VBScript、JavaScript等,它们简单易懂,容易掌握。使用脚本语言编写的程序可以嵌在多媒体文档或应用中,从而方便地实现用户定制的功能。

多媒体著作工具作为一类特定用途的程序,并没有具体的设计标准。目前可见到的千百种著作工具的著作方式多种多样,但归纳起来可分为下列3类:基于流程图的、基于卡片的和基于时间的。①基于流程图的工具功能强大,但要求用户有相当的程序设计经验。Macromedia Authorware(现归属于Adobe公司)是此类著作工具的代表,它采用一种解释型的基于流程的图形编程语言,通过图标的调用来编辑流程图用以替代传统的高级语言编程。②基于卡片的著作工具是按照超链接的结构设计的。超链接的节点由具有一定时空关系的多媒体数据构成,通常被看作卡片、页或场景,用户可直观地编辑卡片内的多媒体内容,操作直观而简便。Asymetrix公司的Toolbook和微软公司的PowerPoint就属于这种类型。③基于时间的著作工具以可视的时间轴来决定事件的顺序和持续时段,它支持多轨处理,可以同时激活多个媒体对象,还可以控制节目的跳转,以增加导航和交互控制能力。Adobe公司的Director和Flash是其典型代表。

多媒体著作工具生成的多媒体文档可以是私有的(自定义的文档格式),也可以是开放的,即遵循已有的公开的多媒体文档规范。

参考文献

1. Fenrich P. Practical guidelines for creating instructional multimedia applications. Dryden Press, 1997

2. http://springer. metapress. com/content/wk4413007231m01k/fulltext. html (徐光祐 张福炎)

duomotai renji jiaohu

多模态人机交互(multimodal human-computer interaction) 通过对视频、音频、图像、触摸、笔等多种模态信息的综合处理,为用户提供自然、直观的人机交互方式的技术。多模态可以理解为多通道或多方式,是相对于键盘、鼠标、显示器等传统的单一的交互方式而言。

人与人之间自然的交互通过视觉、听觉、触觉、嗅觉甚至味觉等多种模态信息来完成。与此相对,

目前的计算机用户主要使用键盘和鼠标等人工设备进行人与计算机的交互,这不利于使更多的非计算机专业人员使用计算机。多模态人机交互研究的目的是使计算机具有检测、感知和理解多种信息的能力,从而使用户能像人与人之间那样与计算机交互。

为使计算机具有感知能力,计算机或计算设备需要装备各种传感器和相应的识别、理解技术作为其输入系统,同时还需要装备相应的多模态输出系统。对于视觉功能,需要装备摄像机,研究计算机视觉技术、人的情感理解、人的运动跟踪等。对于听觉功能,计算机需要装备拾音器,研究言语识别、言语合成、文字到言语转换的**文语转换**技术等。对于触觉需要有数据手套、操纵杆、振动反馈器等触觉传感器和相应的信号处理和理解技术。

基于动态上下文的多模态信息的处理和理解是多模态人机交互的重要的研究方向。上下文包括交互设备、交互环境、交互历史以及交互的任务等所有与人、机、交互及环境相关的信息。基于动态上下文的人体动作、行为理解,是多模态人机交互研究的热点之一,其中包括基于加速度传感器和陀螺仪的手势、人体运动分析与理解,基于视觉设备的脸部表情、注视方向和位置、头和身体的位置和姿态、人体运动的跟踪和理解等。在基于生物特征的身份识别中已出现把多种行为(言语识别、手写体识别、人体运动识别)与生理特征(指纹、视网膜识别)利用传感器融合技术组合起来的趋势。

在人机接口中采用多模态人机交互技术还具有以下优点:①使不同模态的信息能相互补充和支持,从而使系统的性能比用单模态信息更为可靠。例如,把基于视觉信息的唇读识别与言语识别相结合,在高噪声环境下会显著提高单纯利用言语识别系统的性能。②使用户能根据变化的工作环境改变或切换不同的模式。

由于未来用户的终端设备已不只是桌面计算机,而将越来越多地使用笔记本电脑、PDA、移动电话等便携和手持计算设备,其中多数设备没有键盘和鼠标,这使得多模态交互技术如笔输入、言语识别或其他的感知输入就成为必要。

近年来人-计算机交互已扩展到人-机-人交互,使计算设备支持或辅助人与人之间的交互,例如,远程会议与交流、协同工作等,最新发展出来的用于残疾人的脑机接口可以使丧失说话和运动能力的残疾人实现与人的交流。这样,多模态人机交互技术就有了更为广泛的应用领域。

参考文献

1. 徐光祐,陶霖密,张大鹏,史元春. 物理空间与信息空间的对偶关系. 科学通报, 2006, 51(5): 610-616

2. 陶霖密. 情感计算研究进展与展望. 中国图象图形学报,2009, 14(5): F02

3. 董士海,王坚,戴国忠,等. 人机交互和多通道用户界面. 北京: 科学出版社,1999

4. Oviatt S. Multimodal interface. In: Jacko J, Sears A, eds. The Human-Computer Interaction Handbook, Lawrence Erlbaum, 2002 (陶霖密　史元春)

duomotai shengwu tezheng ronghe

多模态生物特征融合(multi-modal fusion in biometrics)　在辨认个人身份时综合考虑多种生物特征以提高识别性能的技术。

单个生物特征识别技术在鲁棒性、稳定性等方面还存在一定的问题。比如在噪声环境中利用语音的说话人识别系统将不能很好地运行(参见**说话人识别**);对于双胞胎,仅仅使用脸像特征也不能很好地区分(参见**人脸识别**);而对于**指纹识别**技术,有5%左右的人不能得到很好的指纹特征(参见**指纹识别**)等。多模态生物特征融合技术,将多种生物特征联合以试图解决上述问题。比如考虑脸像、语音、指纹等多种生物特征的融合,当在噪声环境中语音特征不能很好发挥作用时,其他两个生物特征仍然能够得到很好的辨识效果。

多模态生物特征融合的研究内容主要包括:

(1) 融合的模态问题。即使用哪些特征加以融合。事实上,几乎所有的生物特征都可以用来加以融合,不同的生物特征从不同的侧面反映了人的特性,因此将其通过一定的策略融合后均可以得到比单个特征更好的性能。语音和视觉特征由于说话时的必然相关性因而成为融合的首选。也有研究把指纹、脸像、语音等加以融合。图1给出了基于语音视觉特征的融合示意图,其中视觉特征又包括脸像和唇动两个方面。

(2) 融合的层级问题。不同的生物特征可以在不同的级别上融合,例如:①数据级融合:对最底层的原始数据直接进行融合,因而包含了对原始特征最充分、最有效的描述。然而由于数据间的强相关性、特征的复杂性等,使得该层级只能是理论上的存在,离实际应用还有较大的距离;②参数级融合:具有关联关系不同的特征在参数提取及计算时往往

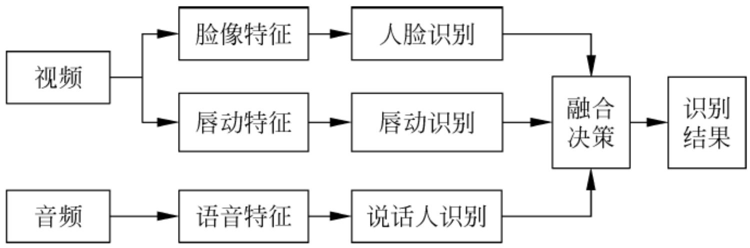


图 1　基于脸像、唇动、语音的决策融合示意图

具有相关性,通过参数级的融合,可以在计算某一特征参数的同时利用另一特征提供的信息;③特征级融合:输入数据经过前端处理后,分别得到每个生物特征的特征描述向量,在特征级融合阶段,将多个低维特征向量融合(合并)成高维的联合特征向量参数,并针对该高维特征向量建立模型,进行识别和决策;④模型级融合:考虑多个生物特征的关系,在此基础上建立联合模型,并基于此模型进行分类决策,在建立联合模型时,既可以考虑特征的关联,也可以考虑其区别特性;⑤决策级融合:具有不同模态的特征分别进行单模态的建模与识别,并将各自识别的中间结果参数,通过决策融合模块进行融合,最后通过多模态决策算法得到最终的鉴别结果。

上述各个层级融合的关系示意如图 2 所示。

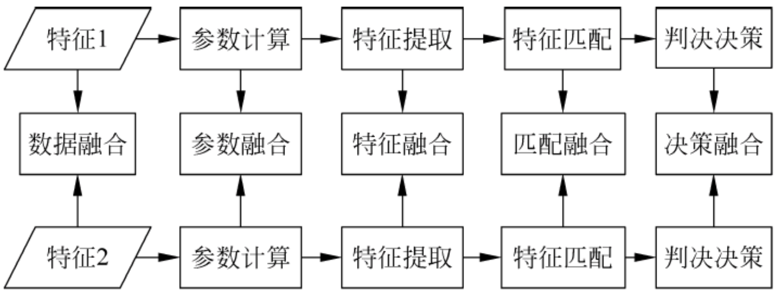


图 2　多层级融合框架结构示意图

目前数据融合的研究主要集中在决策融合阶段,其优点是简单可行,不同模态的单个特征可以分别独立处理,得到单特征匹配结果,然后通过决策融合算法加以综合得到最终的结果,而且可以方便地进行特征的扩充或删减,而不会对整个系统的结构产生影响,参见图 1。但是其缺点也是明显的,不同特征之间完全独立,忽视了特征之间的关联关系所带来的作用和影响。解决这个问题需要在较低层级,如参数级、特征级等层面上,展开更深入的融合研究。

(3) 融合的策略问题。数据融合需要通过一定的策略与数学方法将特征和数据加以综合,这些策略和方法的研究也是目前受到普遍关注的问题。关于数据融合与决策目前有多种算法:决策树、线性判别函数、线性分类器、贝叶斯决策、K 近邻(KNN)、支持向量机(SVM)等。

参考文献

1. Ross A, Jain A, Qian J. Information fusion in biometrics. In: Proc of 3rd Int'l Conference on Audio-and Video-Based Person Authentication, Sweden, June 6—8, 2001

2. Frischholz R, Dieckmann U. BioID: a multi-modal biometric identification system. Computer, 2000, 33(2): 64-68

3. Chibelushi C, Deravi F, Mason J. A review of speech-based biomodal recognition. IEEE Trans on Multimedia, 2002, 4(2): 23-27（蔡莲红　吴志勇）

duotai leixing

多态类型(polymorphic type)　多态类型及其演算研究始于 J. Y. Girard 的关于二阶直觉主义逻辑的类型系统和 J. Reynolds 的多态程序的类型系统。多态类型的语法理论有高阶和二阶之分。高阶多态类型分两层次:先定义种属和构造子,后定义类型。一个项的种属、类型依赖于所含变元的种属、类型。因此种属、类型的说明都涉及它们与变元的联系。归纳地定义有效种属:T,*Type* 都为种属,若 K_1,K_2 为种属,则 $K_1 \times K_2$(序偶),$K_1 \to K_2$(抽象)也都为种

属。种属环境为构造子变元连同其规定的种属组成的有限集 $\Delta=[x_1: K_1,\cdots,x_n: K_n]$，其中基本表示"$x_i: K_i$"可读成变元 x_i 有种属 K_i。有效构造子是指构造子变元被环境所说明者。

种属规则（规则陈述如何赋种属）：若 Δ 为种属环境。

(1) 若 $x: K\in\Delta$，则 $\Delta\vdash x: K$；（可读成：环境 Δ 中对变元 x 赋种属 K，以下类同）

(2) $*: T$；

(3) 若 $\Delta\vdash C_i: K_i(i=1,2)$，则 $\Delta\vdash\langle C_1,C_2\rangle: K_1\times K_2$（序偶）；

(4) 若 $\Delta,x: K_1\vdash C: K_2$，则 $\Delta\vdash\lambda x: K_1. C: K_1\rightarrow K_2$（抽象）；

(5) 若 $\Delta\vdash F: K_1\rightarrow K_2$，$\Delta\vdash C: K_1$，则 $\Delta\vdash FC: K_2$（作用）。

类型定义为种属是 $Type$ 的构造子（Δ 为种属环境）：

(1) 若 $X: Type\in\Delta$，则 $\Delta\vdash X: Type$；

(2) $1: Type$；

(3) 若 $\Delta\vdash A: Type$，$\Delta\vdash B: Type$，则 $\Delta\vdash A\times B: Type$（序偶），$\Delta\vdash A\rightarrow B: Type$（抽象）；

(4) 若 $\Delta,x: K\vdash A: Type$，则 $\Delta\vdash\forall x: K. A: Type$。

类型环境是自由变元连同规定的类型组成的集合 $\Gamma=[X_1: A_1,\cdots,X_n: A_n]$。若 $\Delta\vdash A_1,\cdots,A_n$，则 Γ 称为在 Δ 之下良定义。

类型的规则（陈述如何在 Γ 中赋类型）：Γ 为 Δ 之下良定义。

(1) 若 $x: A\in\Gamma$，则 $\Gamma\vdash_\Delta x: A$；

(2) $\Gamma\vdash_\Delta *: 1$；

(3) 若 $\Gamma\vdash_\Delta a_i: A_i(i=1,2)$，则 $\Gamma\vdash_\Delta\langle a_1,a_2\rangle: A_1\times A_2$；

(4) 若 $\Gamma\vdash c: A_1\times A_2$，则 $\Gamma\vdash\Pi_i c: A_i(i=1,2)$（分量）；

(5) 若 $\Gamma,x: A\vdash_\Delta b: B$，则 $\Gamma\vdash_\Delta\lambda x: A. b: A\rightarrow B$；

(6) 若 $\Gamma\vdash_\Delta f: A\rightarrow B$，$\Gamma\vdash_\Delta a: A$，则 $\Gamma\vdash_\Delta fa: B$；

(7) 若 $\Gamma\vdash_{\Delta x: k} c: C$，$x$ 不出现于 Γ 中类型，则 $\Gamma\vdash_\Delta\Lambda x: K. c: \forall x: K. C$；

(8) 若 $\Gamma\vdash_\Delta F: \forall x: K. C$，$\Gamma\vdash D: K$，则 $\Gamma\vdash_\Delta FD: C[D/X]$。

归约等式定义为：

(1) $(\Lambda X: K. F)C\overset{\beta 2}{=}F[C/X]$；

(2) $\Lambda X: K. FX\overset{\eta 2}{=}F$；

(3) $(\lambda x: A. f)a\overset{\beta}{=}F[a/x]$；

(4) $\lambda x: A. fx\overset{\eta}{=}f$；

(5) $\Pi_1\langle t_1,t_2\rangle\overset{\Pi_1}{=}t_1$；

(6) $\Pi_2\langle t_1,t_2\rangle\overset{\Pi_2}{=}t_2$；

(7) $\langle\Pi_1 t_1,\Pi_2 t_2\rangle\overset{\sigma}{=}t$；

(8) $u\overset{!}{=}*$。

其中 u 为 T 或 1，t 为类型 $A_1\times A_2$，或为种属 $K_1\times K_2$，以上为高阶演算。二阶演算只是省略种属直接考虑类型 $Type$。多态类型演算具有丘奇–罗瑟性质和强典范性质。基于多态类型演算可定义二阶直觉主义逻辑：

$P\rightarrow Q$	定义为	$P\rightarrow Q$
$\top$	定义为	$\forall X: Type. X\rightarrow X$
$\bot$	定义为	$\forall X: Type. X$
$P\wedge Q$	定义为	$\forall X: Type. (P\rightarrow Q\rightarrow X)\rightarrow X$
$P\vee Q$	定义为	$\forall X: Type. (P\rightarrow X)\rightarrow(Q\rightarrow X)$
$\neg P$	定义为	$P\rightarrow\bot$
$\exists_{x: A}. P[X]$	定义为	$\forall X: Type. (\forall x: A(P[X]\rightarrow X))\rightarrow X$

参考文献

1. Girard J Y. Interprétation fonctionelle et Élimination des Coupures Dans Lárithmétique Dórdre Supérieure. Ph. D Thesis, Université Paris Ⅶ, 1992

2. Reynolds J. Polymorphism is not Set-Theoretic. In: Kaha G, et al, eds. Semantics of Data Types. Springer-Verlag, 1984: 145-156

3. Reynolds J, Plotkin G. On functors expressible in the polymorphic typed lambda calculus. LFCS Report Series 53, University of Edinburgh, May, 1988

（陆汝占）

duotongdao touying xianshi

多通道投影显示（multiple projection displays）　将多台投影仪的投影画面进行拼接处理，在平面、曲面或不规则表面上呈现出一个完整、无畸变、无色差显示画面的显示技术。利用多通道投影显示技术，可以克服单台投影仪在投影范围、投影景深、投影解析度、投影亮度等多方面的局限，实现大范围、复杂景深、高分辨率、高亮度的投影内容呈现。

多通道投影显示技术已广泛应用于大型文博科技场馆展览展示、虚拟战场仿真、数字城市、科学计算可视化、数字娱乐等领域。

为实现多台投影仪投影画面的拼接显示,多通道投影显示技术包含几何校正、颜色校正、重叠区域融合、同步显示等多项技术。

1. 几何校正

为了实现用多台投影仪显示画面的拼接,需要弥补由各台投影仪因投影仪摆放位置、投影角度以及投影仪镜头的差异在拼接显示时造成拼接画面在几何上的错位,而几何校正就是计算并校正该差异的技术与方法。其基本原理是在用多台投影仪进行画面显示时,将待显示画面做某种形式的分割,形成与投影仪数目对应的若干小幅子画面,然后在每个小幅子画面传输到对应投影仪进行显示之前,对子画面进行几何预扭曲,使得最终投影在屏幕上的多幅子画面能够拼接出一幅完整且无几何形状畸变的画面。因此,几何校正的核心是得到对画面的几何预扭曲变换。

在几何校正技术上,可以根据投影平面的类型,分为针对平面、针对二次曲面、针对自由曲面以及针对不规则曲面的几何校正技术。在具体实施几何校正时,又可以分为两种:①采用人机交互的手动校正　通过提供一个所见即所得的图像扭曲软件工具,用户一边对待显示画面进行预扭曲,一边将扭曲后的画面直接通过多投影系统显示出来,由用户来不断地调整画面扭曲方式和程度,使得显示出的画面达到令用户满意的程度;②采用摄像机或照相机的自动校正　通过摄像机或照相机来代替人眼观测屏幕表面形状,再采用适当的算法自动计算出几何预扭曲变换。

2. 颜色校正

由于光电元器件特性的个体差异,不同投影机在成像上会出现颜色差异。在多投影显示中为了实现颜色一致性,就需要弥补这种颜色差异。颜色校正就是计算并校正该差异的技术。一般颜色校正有两种方式,一种方式是设法调整投影机本身的硬件参数,以保证光学成像系统的一致性;另一种方式是在将子图像传输给投影机之前,对每幅子图像的每个像素颜色根据投影机的显示颜色特征进行校正映射,使得最终投影在屏幕上的多幅子图像的显示颜色满足一致性要求。后一种方法需要对每台投影机的显示颜色特征进行测量。这种测量往往需要借助光度学的测量仪器和方法。现在也有采用高动态图像的摄影测量方式来近似获得。

3. 重叠区域颜色融合

在多台投影机进行拼接时,由于投影区域的重叠,会造成部分重叠区域颜色过亮,需要采用重叠区域颜色融合技术来消除投影重叠区域的高亮,使融合后的投影画面在重叠区域上过渡自然。重叠区域颜色融合一般采用两种技术方案:一是通过在投影仪上设置光线遮罩,利用光学器件的遮挡效应,实现颜色的正常过渡;二是通过软件设置的方法,在显示图像上设置混合模板,控制投影机在重叠区域的颜色输出来实现重叠区域的颜色过渡。在进行颜色重叠区域过渡设置中,又可分为最低过渡颜色设置、梯度连续过渡颜色设置等方法。

4. 同步显示

在多台投影机进行拼接时,为保证各个子画面在时间上的一致性,需要进行同步显示。同步显示就是利用与投影机连接的计算机,控制投影机显示画面的速度与频率,实现多台投影机同步的画面显示。实现同步显示的方法一般有两种,一是通过设置单一的信号触发源发出固定频率的触发信号,控制显示的计算机通过接收触发信号实现显示内容的同步;另一种是利用网络,通过由服务器发送帧同步数据包,各个与投影机相连的计算机通过接收与反馈帧同步数据包来实现子画面的同步输出。前者由于采用专有设备,同步显示误差小,适合对同步显示有严格要求的应用。后者基于现有网络环境,部署方便,成本较低,但同步精度取决于网络性能。不过随着网络性能的提高,其同步精度已能满足绝大多数场合的需要。

参考文献

1. Majumder A, Brown M S. Practical Multi-Projector Display Design. A. K. Peters, 2007

2. Wallace G, Anshus O J, Bi P, et al. Tools and Applications for Large-Scale Display Walls. IEEE Computer Graphics and Applications, 2005, 25(4): 24-33

(华炜)

duoxiangshi kongjian guiyue

多项式空间归约(polynomial space reduction)　一种特殊的、归约函数在多项式空间可计算的**复杂性归约**。

设 L_1, L_2 是 Σ 上的两个语言,若存在函数 S: $N \to N$,及 S 空间可计算函数 f: $\Sigma^* \to \Sigma^*$,使得

（1）对任何 $x \in \Sigma^*$，$x \in L_1$ 当且仅当 $f(x) \in L_2$；

（2）存在正整数 C，使对任何 $x \in \Sigma^*$ 有 $S(|f(x)|) \leqslant CS(|x|)$，则称 L_1 可 S-空间归约到 L_2，记为 $L_1 \leqslant_m^S L_2$。特别当限制 S 为多项式函数时，则称 L_1 可多项式空间归约到 L_2。空间归约中较为重要的一种是限制 S 为对数函数 log，称之为对数空间归约，可对复杂性类进行更“细”的划分，特别是研究 P 和 N log 等复杂性类，其时间资源不超过确定的多项式时间时，多项式时间归约则无法对其进行分类。

参考文献

Balázar J L, Díaz J, Gabarró J. Structural complexity Ⅰ. Berlin: Springer-Verlay, 1988

（马绍汉　李大兴）

duoxiangshi puxi

多项式谱系（polynomial hierarchy）　递归论中克林算术谱系的多项式变形。很多似乎不在 NP 类中的计算问题属于多项式谱系的某一层次。多项式谱系的基本思想是 R. Karp 于 1972 年提出的，A. Meyer 和 L. Stockmeyer 在 1973 年给出了多项式谱系的严格形式化定义。

基于多项式时间图灵归约和多项式时间非确定图灵归约的概念，可建立 P 和 NP 类关于任何语言 L 的相对化定义，它们分别记为 P(L) 和 NP(L)，有

$$\mathrm{P}(L) = \{L' \subseteq \Sigma^* \mid L' \leqslant_T^{\mathrm{P}} L\}$$

$$\mathrm{NP}(L) = \{L' \subseteq \Sigma^* \mid L' \leqslant_T^{\mathrm{NP}} L\}$$

这种对 P 和 NP 类关于语言的相对化概念，可自然地推广到任何语言类 $\mathscr{C}$ 上：

$$\mathrm{P}(\mathscr{C}) = \bigcup_{L \in C} \mathrm{P}(L),\ \mathrm{NP}(\mathscr{C}) = \bigcup_{L \in \mathscr{C}} \mathrm{NP}(L)$$

基于这种定义，可将 P 和 NP 视为语言类上的一种算子，且有 $\mathscr{C} \subseteq \mathrm{P}(\mathscr{C}) \subseteq \mathrm{NP}(\mathscr{C})$，$\mathrm{P}(\mathrm{P}) = \mathrm{P}$，$\mathrm{NP}(\mathrm{P}) = \mathrm{NP}$，从语言类 P 开始，将算子 NP 重复地作用在其上，便产生一个语言类的无穷递增序列：

P，NP，NP(NP)，NP(NP(NP))，…

它们依次记为 $\Sigma_0^P, \Sigma_1^P, \Sigma_2^P, \Sigma_3^P, \cdots$，即

$$\Sigma_0^P = \mathrm{P},\quad \Sigma_{k+1}^P = \mathrm{NP}(\Sigma_k^P),\quad k \geqslant 0$$

另外，还可定义两类与 Σ_k^P 相关的复杂性类 Π_k^P 和 Δ_k^P：

$$\Pi_k^P = C_0 - \Sigma_k^P = \{L \subseteq \Sigma^* \mid \bar{L} \in \Sigma_k^P\}$$

$$\Delta_0^P = P,\ \Delta_{k+1}^P = P(\Sigma_k^P),\quad k \geqslant 0$$

这三种复杂性类有下述基本关系：

$$\Delta_k^P \subseteq \Sigma_k^P \cap \Pi_k^P,\quad \Sigma_k^P \cup \Pi_k^P \subseteq \Delta_{k+1}^P$$

由此可见

$$\bigcup_{k \geqslant 0} \Sigma_k^P = \bigcup_{k \geqslant 0} \Pi_k^P = \bigcup_{k \geqslant 0} \Delta_k^P$$

由 Σ_k^P, Π_k^P 及 Δ_k^P（$k \geqslant 0$）所描述的层次结构记为 PH，并称 PH 为多项式谱系。

多项式谱系也可如同算术谱系那样，用交替量词的形式来表示。两者之间的区别仅仅是存在量词 $\exists y$ 代之以多项式存在量词 $\exists^P y$；全称量词 $\forall y$ 代之以多项式规模全称量词 $\forall^P y$；递归集（语言）代之以多项式时间可计算语言。这就是 C. Wrathall 定理：对于所有 $k \geqslant 0$，

（1）$L \in \Sigma_k^P$ 当且仅当存在 $L' \in P$，使得 $x \in L$ 当且仅当 $\exists^P y_1 \forall^P y_2 \cdots Q y_k \langle x, y_1, \cdots, y_k \rangle \in L$。其中当 k 为偶数时，Qy_k 为 $\forall^P y_k$；当 k 为奇数时，Qy_k 为 $\exists^P y_k$。

（2）$L \in \Pi_k^P$ 当且仅当存在 $L' \in P$，使得 $x \in L$ 当且仅当 $\forall^P y_1 \exists^P y_2 \cdots Q' y_k \langle x, y_1, y_2, \cdots, y_k \rangle \in L'$。其中当 k 为偶数时 $Q' y_k$ 为 $\exists^P y_k$；当 k 为奇数时，$Q' y_k$ 为 $\forall^P y_k$。

在 Wrathall 定理中的 $\exists^P y$ 意指存在多项式 P，对于满足 $|y| \leqslant P(|x|)$ 的某些 $y \in \Sigma^*$；$\forall^P y$ 意指存在多项式 P，对于满足 $|y| \leqslant P(|x|)$ 的所有 $y \in \Sigma^*$。

多项式谱系的这种表述形式，对于分析计算问题所处的层次常常更为方便。

对于任意 $k \geqslant 0$，令 k-QBF 表示所有以存在量词开始，至多有 k 个交替量词的可满足量化布尔表达式构成的集合；令 k-$\overline{\mathrm{QBF}}$ 表示所有以全称量词开始，至多有 k 个交替量词的可满足量化布尔表达式构成的集合；令 QBF 表示所有可满足的量化布尔表达式的集合。由 Wrathall 定理知，

$$k\text{-QBF} \in \Sigma_k^P,\quad k\text{-}\overline{\mathrm{QBF}} \in \Pi_k^P$$

并由 0-QBF = P，1-QBF = SAT，容易看出，k-QBF 是 Σ_k^P-m-完全的；k-$\overline{\mathrm{QBF}}$ 是 Π_k^P-m-完全的。考虑到 QBF 是 PSPACE-m-完全的这一事实，有

$$\mathrm{NP} \subseteq \mathrm{PH} \subseteq \mathrm{PSPACE}$$

多项式谱系与其他复杂性类，以及多项式谱系自身的各种复杂性类之间的包含关系，究竟是真包含关系还是相等关系，尚待研究。

多项式谱系中一个有趣而重要的未解决问题是其层次是否塌陷。亦即是否存在 $k \geqslant 0$，使 $\Sigma_k^P = \Sigma_{k+1}^P$。不难证明，若 $\Sigma_k^P = \Sigma_{k+1}^P$，则 $\Sigma_k^P = \Pi_k^P = \Sigma_{k+1}^P = \Pi_{k+1}^P = \cdots = \Sigma_{k+j}^P = \Pi_{k+j}^P$，对于任何 $j \geqslant 0$ 成立。从而

推出 $\Sigma_k^p = PH$，亦即多项式谱系塌陷在第 k 层。

由于许多包含在多项式谱系中的复杂性类，都已证明它们属于谱系中的较低层，例如有界误差概率多项式时间类 $BPP \subseteq \Sigma_2 \cap \Pi_2$，因此，许多人认为多项式谱系很可能塌陷在较低层，但未能给出证明。多项式谱系的塌陷问题常与复杂性类的其他问题存在着某些联系，彼此相互影响。

参考文献

1. Garey M R, Johnson D S. Computers and intractability: a guide to the theory of NP-completeness. San Francisco, CA: W. H. Freeman and Company, 1979

2. Balcázar J L, Díaz J, Gabarró J. Structural complexity Ⅰ. Berlin: Springer-Verlay, 1988

（马绍汉　李大兴）

duoxiangshi shijian guiyue

多项式时间归约（polynomial time reduction）　一种常用的、归约函数在多项式时间可计算的**复杂性归约**。S. Cook 于 1971 年利用多项式时间图灵归约，定义了 NP 类中的“最困难”问题。并证明了判别布尔表达式的可满足性问题（SAT），是这类问题的第一个问题。

假设所考虑的问题都已编码成字母表 Σ 上的语言（实例的集合）。设 L_1, L_2 是 Σ 上两个语言，若存在以 L_2 为 oracle 集的多项式时间图灵机 M，其接受的语言为 L_1，则称 L_1 多项式时间图灵归约到 L_2，记为 $L_1 \leqslant_T^p L_2$。这时，对 x 是否属于 L_1 的判别可转化为至多 $|x|$ 的多项式个元素是否属于 L_2 的判别，因此，$L_2 \in p$ 便导致 $L_1 \in p$。从这种相对的意义上讲，L_1 的计算不比 L_2 困难。

$\leqslant_T^p$ 可以是定义在任何语言类 $\mathscr{C}$ 上的一种二元前序关系，如果存在 $L \in \mathscr{C}$，对于任何 $L' \in \mathscr{C}$，都有 $L' \leqslant_T^p L$，则 L 就是 $\mathscr{C}$ 中（在多项式时间图灵归约下）“最困难”的，称其为 $\mathscr{C}T$-完全的。多项式时间图灵归约又称为库克归约。由多项式时间图灵归约的定义，很自然地可产生另一种重要的多项式时间归约，即多项式时间非确定图灵归约。多项式时间图灵归约与多项式时间非确定图灵归约的区别仅在于前者使用的是多项式时间确定型 oracle 机器，后者使用的是多项式时间非确定型 oracle 机器。

R. Karp 于 1972 年利用多项式时间多一归约来刻画 NP 类中的“最困难”问题类。同时，R. Karp 给出了 21 个属于这类问题的实例，它们涉及到逻辑、图论及组合优化等学科中的经典计算问题。对于 Σ 上的两个语言 L_1, L_2，若存在多项式时间可计算函数 $f: \Sigma^* \to \Sigma^*$，使得对任何 $x \in \Sigma^*$，$x \in L_1$ 当且仅当 $f(x) \in L_2$，则称 L_1 多项式时间多一归约到 L_2，记为 $L_1 \leqslant_m^p L_2$。这时，$x \in L_1$ 的判别可以通过计算 $f(x)$，转化成 $f(x) \in L_2$ 的判别。因此，$L_1 \leqslant_m^p L_2$ 更直观地理解为 L_1 的计算不比 L_2 的计算困难。同 $\leqslant_T^p$ 类似讨论，$\leqslant_m^p$ 也可定义在任何语言类 $\mathscr{C}$ 上，若存在 $L \in \mathscr{C}$，使对于任何 $L' \in \mathscr{C}$，都有 $L' \leqslant_m^p L$，则称 L 为 $\mathscr{C}m$-完全的。多项式时间多一归约又称为卡普归约。

递归论中的其他归约都可通过多项式变形成为一种多项式时间归约。上述介绍的几种归约关系已成为计算复杂性理论的重要工具。

参考文献

1. Garey M R, Johnson D S. Computers and intractability: a guide to the theory of NP-completeness. San Francisco, CA: W. H. Freeman and Company, 1979

2. Balcázar J L, Díaz J, Gabarró J. Structural complexity Ⅰ. Berlin: Springer-Verlay, 1988

（马绍汉　李大兴）

duoxieyi biaoji jiaohuan

多协议标记交换（multi-protocol label switching, MPLS）　一种面向连接的可提供高性价比和多业务能力的交换技术。MPLS 技术主要是引入标记交换的概念，通过预先建立的标记交换表进行数据分组的选路和标记交换。MPLS 通过使用显式的标记交换路径 LSP（label switching path），不仅实现了对 IP 实时性业务服务质量 QoS（quality of service）的支持，而且还能够通过对 LSP 的合理部署实现了网络资源的优化以及网络性能的提高等。“多协议”表示在 MPLS 上面可以采取多种不同的协议。MPLS 可使用的多种协议，主要包括链路层的诸多协议，如 PPP、以太网、ATM 以及帧中继等。

MPLS 技术背景

在 20 世纪 80 年代，随着 Internet 的迅速普及，人们开始探索如何提高分组转发速度的方法。此时，出现了一种称之为“交换”的研究思路，即用面向连接的方式取代 IP 的无连接分组交换方式，从而不必使用最长前缀匹配的方法来查找路由表，最终

提升了 IP 数据报的转发速度。为了实现交换,为每个分组分配一个标记(label)。当这些分组到达**交换机**(即标记交换路由器)时,交换机读取每个分组的标记,并用这些标记来检索分组转发表。交换机根据分组的标记进行分组的转发,IETF 已完成了多协议标记交换 MPLS 的系列标准。

基本工作原理

在传统的 IP 网络中,分组每到达一个路由器,都必须查找路由表,并按照"最长前缀匹配"的原则找到下一跳的 IP 地址。当网络很大时,查找含有大量项目的路由表要花费很多的时间。在出现突发性通信量时,往往还会使缓存溢出,从而会引起分组丢失、传输时延增大和服务质量下降等。

MPLS 的一个重要特征就是不用可变长度的 IP 地址前缀来查找转发表中的匹配项目,而是给每一个 IP 数据报打上固定长度"标记",然后对打上标记的 IP 数据报用硬件进行转发,这就使得 IP 数据报转发的过程省去了每到达一个路由器都要上升到第三层应用软件查找路由表的过程,因而 IP 数据报转发的速率就大大加快了。而采用硬件技术对打上标记的 IP 数据包进行转发就称为标记交换。

MPLS 网络中包括若干 MPLS 域,MPLS 域是指该域中有许多彼此相邻的路由器,并且所有的路由器都是支持 MPLS 技术的标记交换路由器 LSR。LSR 同时具有标记交换和路由选择两种功能,标记交换功能是为了快速转发,但是在这之前 LSR 需要使用路由选择功能构造转发表。

MPLS 的基本工作过程:①MPLS 域中的各 LSR 使用专门的标记分配协议 LDP(Label Distribution Protocol)交换报文,并找出和特定标记相对应的路径,即标记交换路径 LSP。②当一个 IP 数据报进入到 MPLS 域时,MPLS 入口节点(ingress node)就给它打上标记,并按照转发表把它转发给下一个 LSR,以后的所有 LSR 都按照标记进行转发。③一个标记仅在两个标记交换路由器 LSR 之间才有意义。分组每经过一个 LSR,LSR 就要做两件事,一是转发,二是更换新的标记,即把入标记更换成出标记。当 IP 数据报进入到下一个 LSR 时,此时的入标记就是刚才得到的出标记。④当 IP 数据报离开 MPLS 域时,MPLS 出口节点(egress node)就把 MPLS 的标记去除,把 IP 数据报交付给非 MPLS 的主机或路由器,以后就按照普通的转发方式进行转发。

MPLS 首部的位置与格式

在把 IP 数据报封装成以太网帧之前,先要插入一个 MPLS 首部。MPLS 首部共包括四个字段:①标记值:占 20 位。②试验:占 3 位,目前保留用作试验。③栈底标记:占 1 位。④生存时间:占 8 位,防止数据分组在 MPLS 域中发生嵌套。

MPLS 还可以使用多个标记,并把这些标记都放入到标记栈(label stack)中。当 MPLS 首部加到 IP 数据报首部的前面时,就可以把这个 MPLS 首部看成是 MPLS 的标记栈。如果再产生一个 MPLS 标记,那么就把它加入到标记栈中,放置在原来标记的前方,即是离 IP 数据报首部更远的位置。MPLS 协议规定,标记栈的栈顶最靠近以太网帧的帧首部,而栈底最靠近 IP 首部。MPLS 标记栈主要用于 MPLS 域出现嵌套的情况。

参考文献

1. 谢希仁. 计算机网络. 5 版. 北京:电子工业出版社, 2007

2. 徐荣, 龚倩, 邓春胜, 田沛. 电信级以太网. 北京:人民邮电出版社, 2009

(张健)

duoxinpian mokuai

多芯片模块(multichip module) 将多个裸露的集成电路芯片直接安装与互连在一块多层高密度的基板上,再用管壳密封制成的一种多功能微电子器件。它具有组装密度高、电气性能好、延迟时间短、体积和功耗小等特点。多芯片模块由芯片、多层基板和封装外壳三部分组成,其基本技术包括裸芯片的微焊接技术、多层基板和封装技术等。

用于多芯片模块的微焊接技术主要有载带自动焊(TAB)(参见**微组装技术**)和倒装焊(FC)(参见**倒装芯片焊接**)两种。

用于多芯片模块的基板材料主要有陶瓷基板(Al_2O_3,AlN,BeO 等)以及硅基板、低温共烧玻璃陶瓷基板(LTCC)、金属基板和金属芯基板等。所用介质材料有二氧化硅、聚酰亚胺等,其热膨胀系数应与硅的热膨胀系数接近,并应具有高的导热系数、低的介电常数和介质损耗系数等。

多芯片模块按其结构与工艺特点可分为下列 3 类:①采用高密度多层印制电路板的 MCM-L 型;②以硅为基板,二氧化硅为介质层,铝或铜为导体的 MCM-Si 型;③采用硅、陶瓷或金属为基板,聚酰亚胺为介质层,铝或铜为导体的 MCM-D 型。MCM-L 采用的是厚膜技术,布线较长,布线板面积也大,限制了速度的提高。MCM-Si 型采用硅基板,导热性能好,且与芯片材料的热膨胀系数相同,因而便于热设

计。MCM-D 的布线电容比 MCM-Si 小,有利于实现高速化,但其导热性能不如 MCM-Si 型。MCM-Si 及 MCM-D 型因其布线工艺采用了光刻技术,有利于提高集成度和高速化,将成为多芯片模块的主要类型。

多芯片模块已成功地应用于**大型计算机**和**巨型计算机**中,美国 IBM 公司曾在 3081 型大型计算机上采用此技术,把 118 个裸集成电路芯片安装互连在 30 层的陶瓷基板上,组成导热模块(TCM)。为了适应日益高速化的要求,目前正以 MCM-D 为中心,开展布线板设计、结构优化设计和改进介质材料性能等研究。随着多芯片模块性能的不断改进和成本的降低,它将被广泛应用于**工作站**、**微型计算机**、通信、医疗电子仪器和汽车电子仪器等领域。

(赵惇殳)

duozhi luoji

多值逻辑(multiple-valued logic) 变元的取值多于两种的逻辑。在通常的逻辑中,命题变元的取值限于真(T 或 1),假(F 或 0)两种,称为 2 值逻辑。例如在“0”,“1”两值之外,增加“u”,以表示非真非假,就得到 3 值逻辑。在其中选取“¬”与“→”作为原始联结词,并规定它们的赋值如下:

A	$\neg A$
0	1
1	0
u	u

$A\to B$	0	1	u
0	1	1	1
1	0	1	u
u	u	1	1

其他命题联结词(∨,∧,↔)则分别定义为:

$$A\vee B=(A\to B)\to B$$

$$A\wedge B=\neg(\neg A\vee\neg B)$$

$$A\leftrightarrow B=(A\to B)\wedge(B\to A)$$

根据上述定义,可以得出这些联结词的赋值表:

$A\vee B$	0	1	u
0	0	1	u
1	1	1	1
u	u	1	u

$A\wedge B$	0	1	u
0	0	0	0
1	0	1	u
u	0	u	u

$A\leftrightarrow B$	0	1	u
0	1	0	u
1	0	1	u
u	u	u	1

如果对 u 作不同的解释,即改变¬与 →的赋值规定,就能得到不同的 3 值逻辑。在上述 3 值逻辑中,每一个恒真命题都是 2 值逻辑中的恒真命题。但反之不然,例如排中律$\neg A\vee A$ 就是最重要的例外。多值逻辑取值的个数并不限于 3,用完全类似的方法,可以构造出 n 值、N_0 值、在[0,1]上取值或在其他适当集合中取值的逻辑。由于多值逻辑演算对于不完善信息提供了可能性推理的工具,故可用于计算机科学、信号系统的设计及量子力学的计算等方面。

设 S 是一个 n 元集合,定义在 S 上而函数值仍属于 S 的函数称为 n 值逻辑函数。当 $n=2$ 时便是 2 值逻辑函数即布尔函数。当 S 是无限集时,便称为无限值逻辑函数。一组 n 值逻辑函数称为完备系或生成系,如果由它们通过迭合运算可生出所有的 n 值逻辑函数,即任意 n 值逻辑函数可由它们表示,若一个函数可生出所有的 n 值逻辑函数,则此函数便称为谢弗函数,例如,$\max(x,y)+1(\mathrm{mod}\ n)$ 便是一个谢弗函数。关于完备系的判定,有著名的完备性定理:一个函数系是完备的必要而且只要该函数系中包含非线性函数、非单调函数、非自对偶函数、非保 $E(<S)$ 函数、非保直接分划函数、非保中心分划函数和非保正则分划函数。每一个函数实际上确定了 S 内的一个代数运算,一个完备系便确定了一个代数系统。常用的代数系统有**波斯特代数**:$\max(x,y)$,$\min(x,y)$,$x+1(\mathrm{mod}\ n)$ 和**模代数**:$x+y$,$x\cdot y$,这里 S 是一个域,相加函数和相乘函数分别是它的加法运算和乘法运算。

多值逻辑函数与多值逻辑网络密切相关,生成系中的每一个函数就是网络中的一个基本元件,不同的生成系就有不同的基本元件。多值逻辑网络的最优设计问题归结为求出多值逻辑函数的最简表达式。因此多值逻辑函数的代数理论是多值数字系统设计和分析的有力数学工具。多值逻辑函数理论广泛用于计算机科学,例如单向函数可作为计算机密码学中的密钥函数。

一般逻辑电路是两个稳定的状态,而多值逻辑电路至少有三个稳定的状态。多值逻辑电路可分为三大类:①电流型电路,以电流值的大小表示逻辑值,典型的技术有集成注入逻辑(I^2L),射极耦合逻辑(ECL)及电流型金属-氧化物-半导体电路(MOS);②电压型电路,以电压值的大小表示逻辑值,典型的技术有多元逻辑电路(NMOS,CMOS 及 DYL);③电荷型电路,以电荷量的大小表示逻辑值,典型的技术有电荷耦合器件(CCD)。

多值逻辑电路与 2 值逻辑电路相比具有较强的逻辑功能,同时能较大地提高集成电路的密度,降低集成电路连接的复杂度,容易解决集成片引线限制问题。因此,多值逻辑电路为超大规模集成电路的发展开辟了新的途径。

参考文献

1. Rine D C. Computer science and multiple-val-

ued logic: theory and applications. Revised ed. Amsterdam: Elsevier Science Publishers, 1984

2. 罗铸楷,胡谋,陈廷槐. 多值逻辑的理论及应用. 北京: 科学出版社,1992

3. 谷超豪. 数学词典. 上海: 上海辞书出版社,1992　（罗铸楷　胡谋）

duo zhinengti xitong

多智能体系统(multi-agent system,MAS)　由多个智能体(Agent)组成的系统。一般情况下,每个智能体具有不完全信息和局部问题求解能力,多个智能体通过合作才能完成任务。多智能体系统(MAS)中的智能体一般指软件智能体,但也可以是机器人甚至可以是人类。MAS 的特征包括: ①每个智能体用于求解问题的信息和能力是不完全的; ②没有系统的全局控制; ③数据是分布的; ④计算是异步的。

按照著名人工智能学者 Minsky 的说法,智能体(Agent)的智能是由一些很小的进程构成,这些小进程只能做一些简单的事情,可能不需要智力和思考。然而当我们把这些小的进程以特定的方式组合在一起,形成了所谓的社会以后,就产生了真正的智能。这些小的进程就是智能体。这显示了怎样从非智能到智能,怎样通过只有很少智力的许多小部分构建出智能的过程。也可以说智能体是指某环境下,具有自治性、反应性、社会性、预动性等的一种或多种行为特征的智能软件或硬件实体。

MAS 主要研究问题求解过程中如何在一群自主的智能体之间进行行为的协调,其中各个智能体之间的关系并不一定是协作的,也可能是竞争甚至是对抗的关系。在 MAS 中可能包含多个局部的问题,甚至其中使用的模型和评价准则都不尽相同。MAS 中的个体可能有相同的结构,也可能是完全异质的,甚至可以是由人和软件智能体组成的团队。

MAS 可以构成一个智能体的社会,其形式包括群体、团队、组织、联盟等。群体形式的 MAS 约束最少,群体中的智能体承担一定的角色,但在交互中不需要相互之间的承诺。团队由为实现某些共同目标而一起工作的、分别承担不同角色的智能体构成,其中个体利益完全服从于整个团队的利益。组织是为完成给定任务建立的,具有更加稳定的结构,由智能体、角色、任务,以及所遵从的规则等要素组成。部分智能体构成的联盟可以完成单个智能体难以完成的任务,或在完成共同任务的同时最大化任务收益。无论哪种形式的 MAS,在合作求解问题中要以智能体之间的彼此交互作为基础。

MAS 吸取了包括经济学、哲学、逻辑学、生态学和社会科学等不同领域的研究成果。但是,MAS 不完全是分布式系统或并发系统,因为智能体在问题求解中的同步和协调不是硬性绑定的,也并非一定有共同的目标,需要通过协商才能达成一致。MAS 也不同于忽略了智能体社会性的早期人工智能,更关注智能的构成(如学习能力、规划、图像理解等),以及多个智能体通过交互形成的群体智能。对策论已经成为分析 MAS 的主要理论工具,但 MAS 不同于经济学和对策论。MAS 关注如何用计算机科学的工具解决问题,而且对策论模型中的理性对于理解人类和智能体并不总是有效和有用的。MAS 提供了为人类社会建模的实验工具,但也不完全是社会科学,通过社会仿真有助于理解复杂的社会现象,但用 MAS 完全准确地模仿人类社会行为是困难的。

MAS 为**软件工程**提供了一种方法和开发范型,是建模、理解和实现以交互为特征的系统的工具。MAS 的应用领域广泛,在工作流和业务过程管理、分布式感知、信息检索和管理、**电子商务**、人机界面、虚拟环境、社会仿真等很多领域得到应用。

参考文献

Wooldridge M. 多 Agent 系统引论. 石纯一,张伟,等译. 北京: 电子工业出版社, 2003　（贺利坚）

duo zhinengti xitong de qiujie fangfa

多智能体系统的求解方法(the solution method in multi-agent system)　研究多个智能体通过合作、协作、协商等交互方式达成一致,共同完成问题求解任务的方法,一般来说所给出的解对参与求解的各方都是有利的。

多智能体系统(MAS)通常处于分布式环境中,一般由异构的、有自身求解目标的智能体构成,采用非集中化设计,没有全局效益约束。在有些情况下,如零和对策(某一方收益的增加必然导致另一方收益的相对应减少,两方收益的代数和为零),可能难以达成一致。在大多数情况下,总能达成对各方有利的一致。一般来说,自治的多智能体求解过程中需要有统一的交互协议和标准,构建这些协议或标准的属性包括: ①Pareto 最优的效用; ②稳定性,没有分离的动机; ③简单性,低计算量和通信量; ④非集中化设计的分布式系统; ⑤对称性,多智能体在求解中扮演基本对称的角色。

当多个智能体联合起来合作求解时，可以完成更大、更复杂的任务，或者能够使其中的每个智能体获得比单独求解更好的收益。多智能体合作求解方法，包括合作、联盟、拍卖、协调、协作和协商等。

合作（Cooperation）　个体与个体、群体与群体之间为达到共同目的，彼此相互配合的一种联合行动方式。合作求解的完整过程分为合作产生、团队初步形成、任务分解、团队最终形成、执行任务等阶段。

一个完整的合作求解模型由群体智能体思维状态模型、合作求解过程以及规划库等部分组成。面对一项不能独立完成的任务，多个智能体首先要形成群体思维状态模型，在此基础上执行合作求解过程，对任务进行分解和分配，并根据已有的规划库，产生合作求解策略，共同实现最终目标。如何准确和客观地描述智能体个体和群体思维状态模型，成为群体智能体合作求解逻辑模型研究的重要内容。

在群体智能体合作求解研究中，也可以运用影响图和贝叶斯图模型的方法描述群体智能体的合作策略，并依此推导出智能体的行为。影响图的应用在很大程度上解决了群体智能体联合策略的表示问题、搜索问题和推理问题。

联盟（Coalition）　是多智能体系统中，由于单个智能体受资源和能力的限制，需要组成联盟以便进行求解。例如现实世界的“拼车”行为就是一种联盟形式，通过多个人组成联盟，从而节省整体和个体的交通费用。

假设存在智能体集合，则一般联盟是该集合的一个真子集。形成联盟时需要对智能体进行不同组合的划分，这种划分称为联盟结构。联盟的研究以对策论、经济学等理论为基础，主要研究内容是①联盟结构的形成，即对智能体集合的划分，以及智能体加入或退出联盟；②联盟求解，以通信、协调、学习、信念修正等方式对联盟的任务、资源、成员进行优化，并完成共同的任务；③联盟收益在成员之间分配，这种分配应满足稳定性、Pareto最优、公平、效用最大化，同时还需考虑计算复杂度和时间成本。联盟结构生成有确定性算法、智能算法和任意时间算法等。

协调（Coordination）和协作（Collaboration）是MAS的两种求解方式，可以使多智能体的知识、意图、规划、行为实现协调甚至达到协作。协作是多个智能体为了共同的目标而采取的共同行动。而如果达到一目标，仅能使其中某一智能体的效用最大，这种情况下智能体间的操作称为协调。

在开放、动态的多智能体系统环境下，具有不同目标的多个智能体须对其目标、资源的使用进行协调。例如，在出现资源冲突时，若没有很好的协调，就有可能出现死锁。而当单个智能体无法独立完成求解目标，需要其他智能体帮助时，则需要协作。

对协调与协作的研究主要是如何使智能体在无目标冲突的情况下互相帮助，特别是应用对策论研究在近似完备知识情况下，智能体在封闭环境中的静态协商规划模型。对策论关于智能体行为协调、协作的研究是以Von Neumann的效用理论为基础的。如果对于任一目标，使某一智能体的效用最大，却不能使另一智能体的效用最大，协调就成为必要的。在实际系统中，协调总是必需的，而协作则不一定总能实现。

协商（Negotiation）　是智能体之间通过妥协就某些问题达成一致而相互有利的方法。协商的内容称为协商议题，有单议题协商和多议题协商。协商方法包括顺序协商、同时协商，以及一揽子交易协商。协商的性质包括均衡点的唯一性、计算复杂度、Pareto最优可能性、达成一致的时间等。协商的关键问题是协商协议、协商均衡目标、协商策略、协商集合。其中协商协议表示协商的基本框架和智能体交互过程中共同遵守的规则；协商均衡目标表示协商达到什么样的均衡目标才终止；协商策略表示智能体为实现目标而采用的基本出价方式；协商集合表示智能体可能提出建议的空间。

对策论方法是协商过程中采用的基本方法。目前的研究热点是多议题协商。多个智能体之间就多个议题产生的各种可能折衷方式可以导致多种协商结果，也导致了多议题协商计算复杂度加大。多议题协商策略对于协商的结果和协商的难度有着直接的影响，对协商策略的基本要求是保证智能体是理性的，至少是Pareto最优。具有学习能力的智能体可以从多次协商中学习较优的协商策略，如学习协商对手的偏好，从而提高协商效果。

MAS的合作求解方法是MAS研究的重点内容，逐渐成为了研究热点，不管是从理论上还是从应用上看，都有很好的发展前景。

拍卖（Auction）　将传统的拍卖方法应用于MAS的求解是自然的，可以用来解决资源分配或任务分配等问题，包括单一物品拍卖、组合拍卖和多属性拍卖等。组合拍卖一次将多个物品同时拍卖。最

简单的组合拍卖是一种暗标最高叫价拍卖方法,采用胜者决定算法 BOB。为减小组合拍卖中的通信量,部分报价方法由买方先将少部分叫价传给卖方,根据计算的需要,再向买方询问必要的叫价。拍卖中可能存在恶意投标问题,如密封投标拍卖中的恶意投标,基于对策论方法可以得出恶意投标的智能体在第 1 价格和第 2 价格密封投标拍卖中的对称贝叶斯 NASH 均衡。

在实际交易时,往往需要考虑物品更多的属性,例如产品提交时间、各种质量参数、售后服务内容等。能够使买卖双方在物品的各种属性上进行协商的拍卖方法称为多属性拍卖。买方的物品属性估价函数和卖方的物品属性成本函数的多样性使这一类问题复杂化。多属性拍卖关注买方和卖方都存在占优策略或贝叶斯最优策略。在贝叶斯均衡下,买卖双方总效用可达到或接近最大。拍卖方法的研究一方面力图找到从模型、效用、策略、效率到鲁棒性、通信量、私有信息保护等方面都比较满意的拍卖方法,另一方面研究拍卖方法中用到的各种算法,如组合拍卖中的胜者决定算法等。单物品单属性拍卖方法的研究已近完善,研究的重点是组合拍卖、双边拍卖和多属性拍卖等。拍卖可以应用在**电子商务**、网络带宽分配、电力调度、网上零售、电子图书馆等大型分布式系统中。

参考文献

1. 石纯一,张伟. 基于 Agent 的计算[M]. 北京:清华大学出版社,2007

2. Wooldridge M. 多 Agent 系统引论. 石纯一, 张伟, 等译. 北京:电子工业出版社, 2003

(童向荣)

E

eyi daima

恶意代码(malicious code) 对人为编写制造的计算机攻击程序的总称,它是以危害计算机和网络信息安全等不良意图为目的、未经授权便干扰或破坏计算机系统或网络功能的可执行的程序代码,其设计目的通常是肆意占用资源、泄漏信息、攻击应用程序、影响计算机系统的正常使用、破坏被感染计算机系统及数据的保密性、完整性和可用性等。它具有可运行性、欺骗性、隐藏性、顽固性、非授权性和破坏性等特点,恶意代码的传播与执行,可对 IT 资产带来危险或不良后果,具有极大的危害性。

恶意代码的行为表现各异,破坏程度千差万别,但基本作用机制大体相同,其整个作用过程一般分为 6 个部分:①侵入系统——是恶意代码实现其恶意目的的必要条件,即恶意代码的传播;②维持或提升现有特权——恶意代码的传播与破坏必须盗用用户或者进程的合法权限才能完成;③隐蔽策略——为了不让用户发现恶意代码已经侵入系统,恶意代码会通过改名、删除源文件或者修改系统的安全策略来隐藏自己;④潜伏——恶意代码侵入系统后,一般是将自己潜伏起来,等待满足一定的条件并具有足够的权限时再发作;⑤破坏——在满足发作条件时,恶意代码就开始实施其后门非法访问、信息窃取、信息泄密、破坏或删除数据、拒绝服务(DoS)、分布式拒绝服务(DDoS)等破坏性活动;⑥重复①至⑤对新的目标实施攻击。

恶意代码的传播途径主要包括 U 盘等可移动媒介、文件共享、网络扫描、对等(P2P)网络、电子邮件、网页、社会工程和系统漏洞等,具体传播手法包括直接利用操作系统或应用程序自身的漏洞、社会工程学,或者两者的结合。如自启动的蠕虫等恶意代码,本身就是软件,这类恶意代码一般通过对操作系统或软件漏洞的直接攻击进行自我传播和扩散。而像病毒、木马等恶意代码,经常通过电子邮件、恶意网站等,利用受害者的心理操纵他们执行不安全的代码。

恶意代码的触发机制主要有:手动执行、社会工程、半自动执行、自动执行、定时炸弹和条件执行等。

恶意代码的自我保护机制主要有加壳、变形、隐藏、伪装、欺骗、模糊变换、干扰代码、反跟踪、反制、反调试、隐蔽通信、窃取、加密这几种方式。

恶意代码的发展历史可以追溯到 1949 年计算机刚刚诞生起,计算机之父冯.诺依曼在《复杂自动机组织论》便定义了病毒的基本概念,这个时代是大型电脑时代,还没有真正意义上的恶意代码出现,只是在概念和实验室研究阶段。这个时代的代表性恶意代码雏形有:1962 年的 Darwin 游戏、1975 年的“Pervade”、1977 年的“磁芯大战 core war”游戏。

到 20 世纪 80 年代,恶意代码以单机木马和病毒为典型代表,并开始了网络木马和病毒的探索。在这个时期,PC 机开始普及,在好奇心的驱使下,开始了大量针对恶意代码的技术探索和研究,发展出了多种形态的恶意代码。代表性恶意代码有:1982 年诞生的最早的引导区病毒“埃尔科克隆者 Elk Cloner”,1986 年诞生的 PC 病毒“大脑病毒 Brain Virus”和 DOS 文件型病毒“魔鬼病毒 VirDem”,1987 年诞生的第一个网络病毒“耶路撒冷 Jerusalem(也叫黑色星期五)”,1988 年诞生的第一个因特网病毒蠕虫“莫里斯蠕虫”。

20 世纪 90 年代,随着计算机的网络化程度逐步提高,网络传播的恶意代码对人们日常生活影响越来越大,恶意代码越来越受到人们的关注。在这个追名为主要目的的时代,各类恶意代码蓬勃发展,呈现快速增长的势头,开始出现僵尸网络等恶意代码新形态。代表性恶意代码有:1990 年的多态病毒“变色龙 Chameleon”,1991 年的复合病毒“变形虫 Amoeba”,1992 年的多态病毒生成器“米开朗基罗 Michelangelo”,1995 年的宏病毒“概念 Concept”,1996 年的 win95 病毒“博扎 Win95. Boza”,1998 年的 CIH 病毒,1999 年的“梅丽莎 Melissa”、Back Orifice 2000 以及拒绝服务和分布式拒绝服务工具 TFN/trin00。

进入 21 世纪,病毒呈现衰退趋势,目标型木马、间谍软件、僵尸程序等恶意软件呈现稳定上升趋势,网络钓鱼谋利形式的恶意代码开始出现。这个时代的恶意代码普遍具有明显的传播速度快、影响面广、

影响范围大和逐利目的明显等特点。恶意软件更加危险和难于发现，其用途更多地是在于为团体（包括国家）谋取经济或政治利益。代表性恶意代码有：2000 年爆发的脚本病毒“爱虫 Loveletter”，2001 年爆发的恶性网络蠕虫“红色代码 Code Red”以及“尼姆达 Nimda”病毒，2003 年爆发的 SQL Slammer 蠕虫和“冲击波 Blaster”蠕虫，2004 年爆发的“震荡波 Sasser”蠕虫病毒并出现了“网络钓鱼 Phishing”，2005 年出现的首款将僵尸网络与大规模电子邮件发送机制整合起来的病毒“麦涛 MyTob”，2008 年发作的“飞客蠕虫 Conficker”造就了最大的僵尸网络，2010 年出现的首个针对工业控制系统的“震网 Stuxnet”蠕虫病毒，2011 年爆发的磁碟机、鬼影 2、灰鸽子、蝗虫军团、扫荡波等恶意代码。

依据恶意代码的传播方式和行为特性，可将其分为**计算机病毒**、**计算机木马**、**计算机蠕虫**、计算机后门、Rootkit、计算机间谍软件、计算机僵尸程序、逻辑炸弹、**网络钓鱼**、浏览器劫持和即时消息攻击等几种常见类型。

恶意代码的分析技术主要包括：静态分析、动态分析、网络特征提取、特征码分析、启发式检测分析、虚拟环境分析、校验和分析、行为分析、沙箱分析、代码仿真分析等。

恶意代码的防范模式主要包括基于主机的检测控制和基于网络的检测控制等。具体而言，要安装安全的操作系统、启用防火墙、做好系统更新与补丁管理、设置安全的账号与口令、安装恶意代码防护系统、加强注册表管理、关注 IE 浏览器安全问题、检查本地木马程序并使用常用的安全工具来进行防范。

恶意代码的发展非常迅速，主要发展趋势有：日趋复杂和完善、传播和感染方式在迅速地演化、种类更模糊、各种手段融合、多平台攻击、发作和流行的时间越来越短、变种速度越来越快、隐藏技术与隐蔽通信技术日新月异、利益驱动越来越多、量身定做、锁定目标进行定向传播。

参考文献

1. 文伟平. 恶意代码机理与防范技术研究. 北京：中国科学院研究生院，2004

2. 沈维. 恶意代码的分析和防治. 上海：上海交通大学，2007

3. 杨永川，黄淑华，魏春光. 边用边学网络安全技术. 北京：机械工业出版社，2010

4. 覃丽芳. 恶意代码动态分析技术的研究与实现. 成都：电子科技大学，2009

（刘宝旭）

erjinzhi suanshu yunsuan

二进制算术运算（binary arithmetic operation） 对两个二进制数所进行的加法、减法、乘法和除法运算。当操作数以定点形式表示时，称为二进制定点加、减、乘、除法运算；当操作数以浮点形式表示时，称为二进制浮点加、减、乘、除法运算。

参加运算的操作数可以用原码、补码或反码表示。原码运算的操作数一般以原码形式存放在存储器或寄存器中，运算结果也以原码表示。同样，补码（或反码）运算的操作数和运算结果一般用补码（或反码）表示，操作数以补码（或反码）形式存放在存储器中。但由于原码加、减法运算比较复杂，因此在某些存储原码的计算机中，当执行加减法运算时，先将操作数转换成补码，然后运算，最后将以补码表示的运算结果转换成原码，再存储起来。直接以原码参与加减法运算的计算机很少见。

定点运算

定点加减法运算

原码加减法运算 加法运算规则：两数同号（符号位相同），执行加法（绝对值相加）；两数异号（符号位不同），执行减法（绝对值相减），并将绝对值大的数作为被减数。运算结果为绝对值，符号位单独处理。

减法运算规则：两数同号，执行减法（绝对值相减），并将绝对值大的数作为被减数；两数异号，执行加法（绝对值相加）。运算结果为绝对值，符号位单独处理。

补码加减法运算 补码（小数）运算规则如下：

$$[X+Y]_{补} = [X]_{补} + [Y]_{补} \quad (\mathrm{mod}\ 2)$$

$$[X-Y]_{补} = [X]_{补} + [-Y]_{补} \quad (\mathrm{mod}\ 2)$$

两个补码数相加，符号位参加运算，两数和的补码等于两数补码之和。

两个补码数相减，可以取减数的负数，进行补码加法运算，结果即为两数之差的补码。

反码加减法运算 反码小数运算规则如下（假设取每个数为双符号位）

$$[X+Y]_{反} = [X]_{反} + [Y]_{反} \quad (\mathrm{mod}\ 4-2^{-n})$$

$$[X-Y]_{反} = [X]_{反} + [-Y]_{反} \quad (\mathrm{mod}\ 4-2^{-n})$$

这里的反码以 $(4-2^{-n})$ 为模，当第一符号位产生进位时，第二符号位有溢出，必须把溢出的进位加到最低位 2^{-n} 上去。

下面举例说明原码、补码和反码 3 种运算方法与步骤（假设用补码、反码表示的数取双符号位）。

例 1 设 $X=0.1010$，$Y=0.1011$，$X-Y$ 的运算过程见表 1。

例 2 设 $X=0.1011$，$Y=0.1010$，$X-Y$ 的运算过程见表 2。

总之，原码加减法运算比较麻烦；反码加减法运算有时需要将最高位产生的进位信号加到最低位（称为循环加），且零值有两种编码；补码加减法运算简便，且零值只有一种编码，因而得到广泛应用。

整数运算规则类同，仅 mod 取值不同（参见**数制**）。

表 1 原码、补码、反码运算（1）

步　骤	原码运算	补码运算	反码运算
1	比较 X，Y 的大小	$[-Y]_{补}=1.0101$	$[-Y]_{反}=1.0100$
2	减法运算（大数减小数） 0.1011 −0.1010 0.0001	$[X]_{补}+[-Y]_{补}$ 00.1010 +11.0101 11.1111（结果）	$[X]_{反}+[-Y]_{反}$ 00.1010 +11.0100 11.1110（结果）
3	确定结果符号为 1， 结果为 1.0001		

表 2 原码、补码、反码运算（2）

步　骤	原码运算	补码运算	反码运算
1	比较 X，Y 的大小	$[-Y]_{补}=1.0110$	$[-Y]_{反}=1.0101$
2	减法运算 0.1011 −0.1010 0.0001	$[X]_{补}+[-Y]_{补}$ 00.1011 +11.0110 00.0001（结果）	$[X]_{反}+[-Y]_{反}$ 00.1011 +11.0101 100.0000
3	确定结果符号为 0， 结果为 0.0001		最高位进位加到最低位 00.0000 +　　　1 00.0001（结果）

定点乘法运算

在 20 世纪 70 年代以前，在仅有加减运算的低档小型计算机或 70 年代末的早期微型计算机中，是用软件（乘法子程序）来实现乘法运算的。

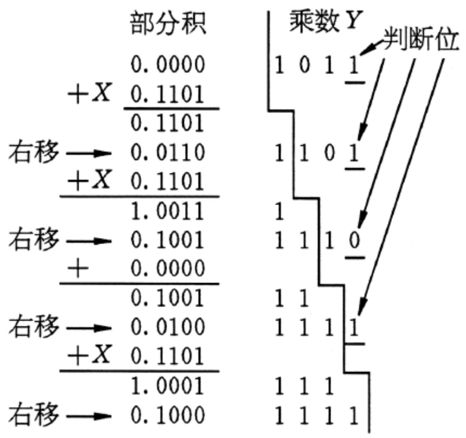
随着集成电路芯片集成度的提高，现在在一般计算机中都有实现乘法运算的硬件，大多与加减法运算共用一个加法器，逐次执行加法与移位操作，分步实现乘法运算。在有些计算机中，为了提高运算速度，设置有专门的乘法器。

常用的定点乘运算方法有原码一位乘法、补码一位乘法、原码两位乘法、补码两位乘法和多位乘法。

原码一位乘法 数值与符号分别处理，两操作数绝对值相乘，符号位相加。两操作数符号相同，乘积为正；符号不同，乘积为负。

执行原码一位乘法时，对应于乘数的每一位（判断位）可得到一项部分积，然后执行 1 次加法和移位操作，当操作数为 n 位时（内含 1 位符号位）执行 $n-1$ 次加法和移位操作。

例 3 设 $X=0.1101$，$Y=1.1011$，求 XY 乘积的过程如下：



符号位 0+1=1

结果(原码) 1.10001111

补码一位乘法　当参加运算的操作数是补码时，可将数转换成原码，然后按原码一位乘法规则运算。也可采用补码一位乘法，直接对补码进行运算，其运算规则如下：

（1）被乘数 $X(X=X_0X_0X_1X_2\cdots X_n)$ 和部分积取双符号位，并参与运算。

（2）乘数 $Y(Y=Y_0Y_1Y_2\cdots Y_n)$ 取单符号位，并在末尾增设附加位 Y_{n+1}，Y_{n+1} 的初始值为0。

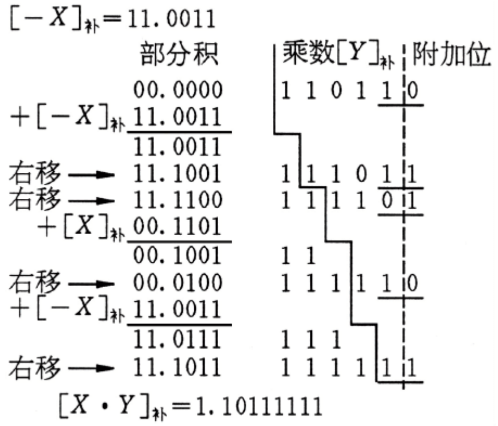
（3）Y_i，Y_{i+1}（第一次为 Y_n，Y_{n+1}）组成各步运算的乘数判断位，其算法如下：

Y_i	Y_{i+1}	操作
0	0	部分积右移1位
0	1	部分积 $+[X]_补$，再右移1位
1	0	部分积 $+[-X]_补$，再右移1位
1	1	部分积右移1位

右移时，最高符号位保持不变。

按上述算法执行 $n+1$ 步，但第 $n+1$ 步不移位。

例4　设 $[X]_补=00.1101$（双符号位），$[Y]_补=1.1011$（单符号位），求 $[X\cdot Y]_补$ 的过程如下：



这种用比较乘数相邻两位来进行乘法操作的方法是由A. Booth夫妇首先提出的，所以又叫***布思算法***。

原码两位乘法　为了提高运算速度，在一次操作中可同时考虑两位乘数，求得与两位乘数相对应的部分积，其速度比一位乘法提高1倍，规则如下：

$Y_iY_{i+1}=00$，相当于 $0\times X$，由于是乘两位，部分积右移两位。

$Y_iY_{i+1}=01$，相当于 $1\times X$，部分积 $+X$，然后右移两位。

$Y_iY_{i+1}=10$，相当于 $2\times X$，部分积 $+2X$，然后右移两位。

$Y_iY_{i+1}=11$，相当于 $3\times X$，因为 $+3X$ 的实现有困难，所以用 $4X-X$ 来代替，在本步中只执行 $-X$，用一个欠账触发器记下欠账 C_j，下一步再补上本步的 $+4X$，由于本步执行 $-X$ 后部分积要右移2位，于是本步的 $+4X$ 操作在下一步只要执行 $+X$ 就可以了。所以原码两位乘法所执行的操作实际上取决于乘数的最低两位 Y_i，Y_{i+1} 和 C_j 的值。

乘法规则如表3所示（$-X$ 用 $+[-X]_补$ 来代替，被乘数与部分积取3个符号位）。

表3　原码两位乘法

C_j	Y_i	Y_{i+1}	操　作
0	0	0	部分积右移2位，置 $C_j=0$
0	0	1	部分积 $+X$，然后右移2位，置 $C_j=0$
0	1	0	部分积 $+2X$，然后右移2位，置 $C_j=0$
0	1	1	部分积 $-X$，然后右移2位，置 $C_j=1$
1	0	0	部分积 $+X$，然后右移2位，置 $C_j=0$
1	0	1	部分积 $+2X$，然后右移2位，置 $C_j=0$
1	1	0	部分积 $-X$，然后右移2位，置 $C_j=1$
1	1	1	部分积右移2位，置 $C_j=1$

补码两位乘法　将补码一位乘法的布思算法与原码两位乘法结合起来，可推导出补码两位乘法的规则。

多位乘法　可在两位乘法的基础上实现多位乘法，或采用阵列乘法器进一步提高运算速度。

定点小数除法运算

根据操作数表示方式的不同，可分为原码除法和补码除法。原码一位除法具体实现时又可采用恢复余数法或加减交替法。为了提高运算速度，还可采用跳0跳1法或迭代法等。

除法运算与乘法运算相似，将 n 位除法操作转换成若干次加减及左移操作，可用硬件或软件实现。

原码一位除法：数值部分相除，符号位相加。现将恢复余数法与加减交替法的运算规则叙述如下。

恢复余数法　被除数减去除数，如果够减（余数为正或0），为溢出；如果不够减（余数为负），商0，并加上除数（恢复余数），被除数左移1位。以后遵循下列规则操作：余数减去除数，如果够减（余数为正或0），商1，余数左移1位；如果不够减（余数为

负),商0,并加上除数(恢复余数),然后余数左移1位。重复执行,直到商满足精度要求为止。当操作数的数值部分为 n 位时,一般重复执行 n 次。

加减交替法(不恢复余数) 被除数减去除数,如果够减(余数为正或0),为溢出;如果不够减(余数为负),商0,余数左移1位,然后遵循下列规则继续操作:如果上次余数为正或0,余数减去除数,得新余数;如果上次余数为负,余数加除数,得新余数。如果新余数为正或0,商1;新余数为负,商0。然后将余数左移1位。重复执行,直到商满足精度要求为止。

浮点运算

浮点数比定点数的表示范围宽,有效精度高,更适合于科学计算和工程计算。

浮点运算可分成规格化和非规格化运算两种,规格化浮点运算的操作数与运算结果都用规格化浮点数表示。其优点是尾数具有较长的有效位数。除了特殊说明以外,一般浮点运算都为规格化运算。

浮点数由阶码和尾数两部分组成。尾数部分的运算与定点数运算方法相似。在不少计算机中,阶码用移码表示,尾数用补码表示,下面的讨论也以此作为依据。

浮点加减法运算

运算步骤如下:

(1) 对阶。两个规格化浮点数的阶码可能不相等,需要将其阶码统一后才能对尾数进行加减法运算。对阶时首先求两数的阶差,然后将阶值小的数的尾数按阶差右移,使两数具有相同的阶码值。

(2) 尾数进行加减法运算。同定点数运算。

(3) 结果**规格化**。尾数运算结果可能是不规格化的。有两种不规格化:第一种是尾数的最高位与符号位相同(即尾数的绝对值小于1/2);第二种是尾数的两个符号位不相同(即尾数溢出,超出了定点小数所能表示的范围)。对于第一种不规格化的运算结果,应将尾数向左移位,每左移一位,阶码减1,直到尾数的最高位与符号位不同为止。这种操作称为向左规格化,简称左规。如果阶码减至小于机器所能表示的数,称为下溢。此时运算结果用机器0表示,即阶码和尾数全为0。对于第二种不规格化的运算结果,应将尾数右移一位,同时阶码加1。这种操作称为向右规格化,简称右规。如果阶码加1后溢出,即运算结果大于机器所能表示的数,称为上溢。

(4) 舍入。在执行对阶或右规时,有可能使尾数的低位移掉,影响数据的精度,为了将移掉的高位保存起来,应采用适当的方法进行舍入。常用的舍入法为0舍1入法(类似于十进制的4舍5入法),如果移掉的最高位为1,则在尾数的末位加1;如果移掉的最高位为0,则舍去移掉的数值。

在尾数末位加1后,有可能使尾数溢出,如果发生这一情况,需右规,阶码加1,并再次判断阶码是否溢出。

浮点乘法运算

运算规则如下:

(1) 阶码相加,如阶码相加后溢出,则表示运算结果溢出。

(2) 尾数相乘,同定点小数乘法。

(3) 向左规格化,并检查下溢。

(4) 舍入,并检查上溢。

浮点除法运算

运算规则如下:

(1) 阶码相减,判溢出(同浮点乘法运算)。并且检查除数是否为0,若是,即为溢出。检查被除数是否为0,若是,结果(商)为0。

(2) 尾数相除,同定点小数除法,但此处不判溢出。

(3) 规格化,如果右规,阶码加1,检查上溢。

(4) 舍入,同浮点乘法运算。

参考文献

1. 李勇,等.计算机原理与设计.修订本.长沙:国防科技大学出版社,1989

2. 王爱英主编.计算机组成与结构.4版.北京:清华大学出版社,2007

(王爱英)

eryuan guanxi

二元关系(binary relation) 两集合的笛卡儿积的子集。

设 A 和 B 是集合,$A\times B$ 是 A 和 B 的笛卡儿积。若 $R\subseteq A\times B$,则称 R 为从 A 到 B 的二元关系,简称关系。当 $\langle a,b\rangle\in R$ 时,常用 aRb 表示。如果 $A=B$,则称 R 为 A 上的关系。

设 R 为从 $A\sim B$ 的关系,则称从 $B\sim A$ 的关系 $\{\langle b,a\rangle \mid \langle a,b\rangle\in R\}$ 为 R 的逆关系,记为 R^{-1}。若 S 为从 $B\sim C$ 的关系,则称从 $A\sim C$ 的关系 $\{\langle a,c\rangle \mid$ 有 $b\in B$ 使 $\langle a,b\rangle\in R$ 且 $\langle b,c>\in S\}$ 为 R 与 S 的合成关系,记为 $R\circ S$。称 A 上的关系 $\{\langle a,a\rangle \mid a\in A\}$ 为 A 的恒等关系,记为 I_A。如果 $R\subseteq A\times B$,显然有 $I_A\circ R=R=R\circ I_B$。设 R 为 A 上的关系,令 $R^0=I_A,R^{k+1}=$

$R^k \circ R, k=0,1,2,\cdots$,则有$(R^{-1})^n=(R^n)^{-1}$,$R^n \circ R^m = R^{n+m}$,$(R^n)^m=R^{nm}$,$n,m\in N$。

二元关系的概念可推广至n元关系($n>2$)(参见**关系代数**)。

设R为A上的一个关系,若对于任意$a\in A$,都有$\langle a, a\rangle \in R$,则称$R$是自反的。若对于任意$a\in A$,都有$\langle a, a\rangle \notin R$,则称$R$是反自反的。若$\langle a, b\rangle \in R$,则$\langle b, a\rangle \in R$,则称$R$是对称的。若$\langle a, b\rangle \in R$且$\langle b, a\rangle \in R$,则$a=b$,则称$R$是反对称的。若$\langle a, b\rangle \in R$且$\langle b, c\rangle \in R$,则$\langle a, c\rangle \in R$,则称$R$是传递的。

例如,自然数集合**N**上的大于或等于关系"≥"$\{\langle n,m\rangle \mid n,m\in N$ 且 $n\geqslant m\}$只具有自反性、反对称性和传递性。设m是大于1的正整数,整数集合**Z**上的"模m同余"关系$R=\{\langle a, b\rangle \mid a,b\in Z$ 且 m 整除 $a-b\}$只具有自反性、对称性和传递性。

一些重要的二元关系如下:

相容关系　若集合A上的关系R是自反的和对称的,则称R为A上的相容关系。

等价关系　若集合A上的关系R是自反的、对称的和传递的,则称R为A上的等价关系,常用"≈"表示。

偏序关系　若集合A上的关系R是自反的、反对称的和传递的,则称R为A上的一个偏序关系,常用"≤"表示,并称$\langle A, \leqslant\rangle$为偏序集。设$\langle A, \leqslant\rangle$为偏序集且$S\subseteq A$。设$a\in S$,若当$x\in S$且$a\leqslant x$(或$x\leqslant a$)时必有$x=a$,则称$a$为$S$的一个极大元(或极小元)。若对每个$x\in S$均有$x\leqslant a$(或$a\leqslant x$),则称$a$为$S$的一个最大元(或最小元)。

显然,S的最大元(最小元)必是极大元(极小元),反之不真;若S有最大元(最小元),则必唯一。

设$b\in A$,若对每个$x\in S$有$b\leqslant x$(或$x\leqslant b$),则称b为S的一个下界(或上界)。设$b\in A$为S的一个下界(或上界),若对S的每个下界(或上界)$c\in A$均有$c\leqslant b$(或$b\leqslant c$),则称b为S的一个下确界(或上确界)。

显然,上确界为所有上界的最小元,下确界为所有下界的最大元。因此若上(下)确界存在,则必唯一。

设$\langle A, \leqslant\rangle$为偏序集,如果满足

(1) 若$a,b\in A$,则必有$a\leqslant b$,或$b\leqslant a$;

(2) 若S为A的非空子集合,则S必有最小元。

则称≤为A上的一个良序关系,并称$\langle A, \leqslant\rangle$为一个良序集。集合$A$上的偏序≤如果仅满足上面的条件(1),则称≤为$A$上的一个线性序(或全序),这时称$\langle A, \leqslant\rangle$为一个线性序集(或全序集,或链)。

关系在计算机科学中有广泛的应用。许多系统的结构问题或者数据交换问题都可以表达为关系,从而能够清晰、精确、简明地描述复杂系统(例如一个网络规范,或者一个软件架构)的结构和性质,为提高软件等的安全性和可靠性提供了坚实的数学基础。

利用关系理论使数据库从网络型、层次型转变成关系型,这样使得逻辑结构更简单,数据易于表示、存储和处理。关系数据模型中表结构的确定与设计、关系操作的数据查询和维护功能的实现、关系分解的无损连接分析、连接依赖等问题都用到二元关系。

针对不完备信息系统中信息处理的问题,可利用相容关系构建模型进行研究。在信息检索中,可以根据一个关键字利用等价关系将文献分成两类,这样不同的关键字就可以得到不同的分类,进而把信息分类以便检索。程序设计中常用的排序,如字典排序、拓扑排序等可以用偏序关系(良序关系)进行描述。

参考文献

1. Clarke E M, Grumberg O, Peled D A. Model checking. Cambridge, MA: The MIT Press, 1999

2. Date C J, Drawen H. Databases, types, and the relational model. 3rd ed. Boston: Addison-Wesley Longman Publishing Co., 2006

3. Kryszkiewicz M. Rough set approach to incomplete information system. Information Sciences, 1998, 12(2): 39-49

4. Knuth D E. The art of computer programming Volume 3: Sorting and Searching. 2nd ed. Boston: Addison-Wesley Longman Publishing Co., 1998

（吴尽昭）

F

fabu dingyue moxing

发布-订阅模型(publish/subscrible model)

一种由发布者、订阅者和事件服务三种软件实体组成的基于事件的分布式通信模型,简称 p/s 模型。p/s 模型已广泛应用于新闻订阅、股票报价、电子商务、电子拍卖、航空交通管制、战场态势感知、协同作战和互联网游戏等分布式软件系统中。

如图 1 所示,p/s 模型中的订阅者通过订阅原语向事件服务注册,表达对特定事件的兴趣,也可以通过取消原语向事件服务取消不再感兴趣的订阅;发布者向事件服务发送新产生的事件;事件服务是订阅者和发布者的中介,一方面负责订阅的存储与管理,另一方面将发布者发来的事件以通知的形式及时发送到感兴趣的订阅者。

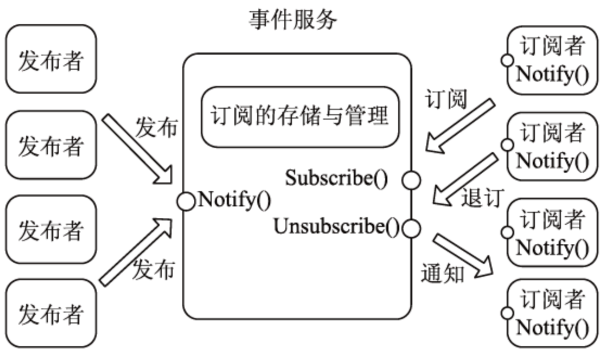


图 1 发布/订阅模型

p/s 模型具有异步、多点通信的特点,参与通信的订阅者和发布者在空间、时间和控制流上是完全解耦的,因而通常能够很好地满足分布式系统松散通信的需求。

p/s 模型的研究内容主要包括事件模型、订阅模型、匹配算法、路由算法、服务质量保障机制和拓扑结构等方面。事件模型定义事件的数据结构,订阅模型定义系统可支持的订阅条件,事件模型和订阅模型共同决定了系统的能力;匹配算法、路由算法和质量保障机制用于系统性能的优化;拓扑结构决定了系统的可扩展性。

按照订阅条件的不同, p/s 模型一般可以分为三类:①基于通道的 p/s 模型 发布者和订阅者都侦听一个通道,发布者产生的事件将发送给侦听该通道的所有订阅者,通道的抽象等同于 IP 组播组地址;②基于主题的 p/s 模型 所有事件都按主题被分成组,每个事件只属于其中一个主题,订阅者按主题给出订阅信息,并接收相应的事件;③基于内容的 p/s 模型 订阅者可以在事件的内容上指定约束条件,表达其感兴趣的事件,这种模型具有更强的表达能力和更好的灵活性,近年来得到了广泛应用。

参考文献

Eugster P T, Felber P A, Guerraoui R, et al. The many face of publish/subscribe. ACM Computing Surveys, 2003, 35(2): 114-131 (魏峻)

fanxiang chuanbo wangluo

反向传播网络(back-propagation network, BPN)

一种多层前向**人工神经网络**模型,因其在学习过程中采用误差反向传播的策略而得名。

误差反向传播的思想最早由 A. E. Bryson 等人于 1969 年提出。1974 年,P. J. Werbos 在哈佛大学的博士论文研究误差反向传播。1985 年 D. Parker 对误差反向传播学习算法作了更进一步研究。直到 1986 年 D. E. Rumelhart 及其研究小组在《Nature》杂志上发表反向传播网络和反向传播算法,实现了 Minsky 的多层前向神经网络的设想,受到许多学者的重视,是人工神经网络发展历史上的一个里程碑。

具有单隐层结构的反向传播网络如图 1 所示。该网络由输入层,隐藏层和输出层组成。给定一组训练样例集,它的工作过程包括两部分:训练阶段和工作阶段。训练阶段由正向信号传播和反向误差传播组成,首先输入模式先经输入层正向传播到隐藏层,经过隐藏层神经元的激励函数后再传播到下一层,在该正向传播的过程中,每一层神经元的状态只影响下一层神经元的状态。如果在输出层不能得到期望的输出,转入执行反向误差传播,即将误差信号沿原来的正向通路返回,用最陡梯度下降法修正各层之间的连接权值与神经元参数,直到满足性能函数的要求。当训练阶段结束后,网络即可实现任意模式的类别划分和输出预测等任务。

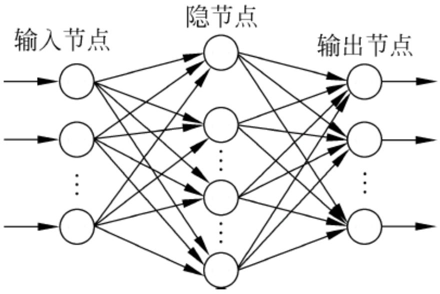


图 1　反向传播网络

反向传播网络是应用最广的人工神经网络之一,在模式分类、模糊控制、函数逼近等领域都得到应用。T. J. Sejnowski 和 C. R. Rcsenberg 利用反向传播网络模型实现了一个英语课文阅读学习机的实验。在这个名为 NetTalk 的系统中,由 203 个神经元组成的输入层把字母发音的时间序列巧妙地变换成空间序列模式,它的中间层(隐藏层)有 80 个神经元,输出层的 26 个神经元分别对应于不同的需要学习的发音记号,并输出连接到由发音记号构成的语音合成装置,构成了一台英语阅读机。实验结果相当成功的,有力地证明了反向传播网络具备很强的学习功能。

反向传播网络仍存在不少问题,网络中隐单元的存在,使梯度下降不能保证全局最小收敛;训练速度太慢;后面样本训练时可能会破坏前面学习的结果;如何设计隐层和隐单元数目等,这些问题尚待进一步研究。

参考文献

1. Hecht-Nielsen R. Theory of the backpropagation neural network. International Joint Conference on Neural Networks. INNS/IEEE, 1989,1: 593-605

2. 史忠植. 神经网络. 北京: 高等教育出版社,2009

(焦李成)

fandaishu

泛代数(universal algebra)　以代数系统为研究对象的**代数学**分支,主要研究内容是不同代数系统的共有的性质。

设 A 是非空集合,n 是正整数,A 的一个 n 元运算 f, 指的是 n 个 A 的笛卡儿乘积 $A\times A\times\cdots\times A$ 到 A 的一个映射。例如实数集 $\mathbf{R}$ 上的加法运算就是 $\mathbf{R}$ 上的一个二元运算,取负元就是 $\mathbf{R}$ 上的一个一元运算。规定 A 的一个空元运算就是在 A 中标定一个元素,也叫常值运算。

为了区别不同的代数系统,有所谓型的概念。两个代数的运算集之间如果存在一一映射,且相对应的运算具有相同的元数,则称两个代数是同型的。例如有单位元的环和布尔代数可看作同型代数,但它们都与群不同型。因为有单位元的环可看成具有两个二元运算(加法和乘法)、一个一元运算(取负元)和两个空元运算(零元和单位元)的代数系统;布尔代数可看成具有两个二元运算(交和并)、一个一元运算(取补元)和两个空元运算(0 和 1)的代数系统;群可看成具有一个二元运算(乘法)、一个一元运算(取逆元)和一个空元运算(单位元)的代数系统。需要特别指出的是,域不能作为泛代数的研究对象,因为对乘法取逆元只对全体非零元成立,不是一元运算,不能把域归于某个型的代数。

设(A,F)是一个代数,θ 是 A 上一个等价关系. 若对 F 中的每个 n 元运算 f, 当 $x_i,y_i\in A$, 使得 $x_i\theta y_i$ $(i=1,2,\cdots,n)$ 时恒有 $f(x_1,x_2,\cdots,x_n)\theta f(y_1,y_2,\cdots,y_n)$, 则称 θ 是 A 上一个同余。这时,若把 A/θ 记为所有等价类的集合,则对 F 中每个 n 元运算 f, 都可以定义 A/θ 上的一个 n 元运算 f_*, 满足条件:对任意的 $a_i\in A(i=1,2,\cdots,n)$, 均有

$$f_*(a_1/\theta,a_2/\theta,\cdots,a_n/\theta)=f(a_1,a_2,\cdots,a_n)/\theta$$

则可获得一个新的代数,称为 A 关于 θ 的商代数。

泛代数是把一般的代数系统作为研究对象,通过深入的分析,研究不同代数系统之间的关系,以及共同的性质,从而建立各种代数系统的"族谱"。在泛代数中,也有同态、同构等基本概念。当一个代数系统的型确定的时候,一般存在三种基本的结构,同态像、子代数、直积。从已知的代数系统出发,可以建立新的代数系统,例如取亚直积、正向极限、反向极限以及自由代数等。在泛代数中,如果可以通过等式型公理来定义一个代数结构,则该代数结构称为一个簇。伯克霍夫定理断言:一个代数构成的类是一个簇当且仅当其中任意代数关于同态像、子代数和直积封闭。

泛代数在计算机科学等学科中得到了广泛的应用,并获得了许多深刻的结果。例如,形式语言和自动机就是两类不同的代数系统,通过对它们形式化的描述和刻画,以及深入的分析,建立了关于自动机和形式语言之间十分精彩的对应关系,为形式语言的编译技术奠定了可靠的理论基础。又例如,计算机内部的进程可以定义为一个代数系统,用所谓的进程代数来进行刻画。通过代数系统之间同态或者同构的揭示,把表面上看来极不相同的系统建立深

刻的关联,进而获得深刻的研究结果。一般来说,任何复杂系统都离不开两类要素。一个称为元素(或者对象),是系统中所有的个体单元;另一类称为动作,这是施加于对象上的操作。从泛代数的角度,对象的集合连同定义在对象上的动作,就构成了组成泛代数系统的基本内容。再根据系统本身的运动规律,附加一些公理刻画,就把实际系统表达为一个代数系统,从而关于代数的结论和方法就可以应用于这个复杂系统的研究。这种把实际系统抽象为代数系统的方法,是计算机科学在解决问题时常用的途径,也为计算机科学提供了有效的研究方法论。

参考文献

Burris S, Sankappanavar H P. A course in universal algebra. New York: Springer-Verlag, 1981

(王利民)

fanzaiwang

泛在网(ubiquitous network) 迄今为止,泛在网并没有严格的定义,通常被理解为是一种广泛存在的网络,它可以在任何时间、任何地点、为任何人及任何物提供顺畅的通信联系。

泛在网概念的产生可以追溯到20世纪末期。1991年,Xerox实验室的计算机科学家Mark Weiser首次提出“泛在运算”的概念,描述了任何人无论何时何地都可通过合适的终端设备与网络进行连接,获取个性化信息服务的全新信息社会。

在此基础上,日韩等国衍生出了泛在网络(ubiquitous network),欧盟提出了环境感知智能(ambient intelligence),北美提出了普适计算(pervasive computing)。这些概念的说法与描述不尽相同,但是观念却相当一致。

这些概念都被视为21世纪的信息通信典范——环境感知智能泛在网络AUN(ambient ubiquity network)的起源。北美、欧洲、亚洲在泛在网技术的发展应用方面有不同的特色,北美侧重于普适计算,欧洲侧重于环境智能感知,而亚洲更重视泛在接入。

环境感知泛在网络AUN具有如下特征:

(1)环境感知性 人们未来的生活周围将出现各式各样的智慧型接口,网络能够感知用户及周边环境场景信息,自动选择合适的传送方式,将正确的服务信息传递给需要的用户。

(2)自组织、自愈性 节点动态、智能地接入网络,并具有分布式管理的功能特点;网络具有高抗干扰、抗故障能力,网络节点具有自恢复能力,传感网络将被广泛应用。

(3)泛在性、异构性 多种接入方式、多种承载方式融合在一起,实现无缝接入;任何对象(人或设备等)无论何时何地都能通过合适的方式获得永久在线的宽带服务,可以随时随地存取所需信息。

(4)开放性、透明性 网络技术对业务及用户体验是透明的,用户无须关心网络状态、网络的接入技术和承载技术。

(5)移动性、宽带性 泛在网络环境的基础是高带宽接入;基于铜缆、光纤的上下行对称宽带接入,基于OFDM、802.1X的固定高速无线接入,基于3G/4G的移动高速数据接入也将被广泛应用。

(6)多媒体、协同性 数字化、多媒体化的信息服务将融入人们日常工作生活中,并起到方便生活、提升效率之功效;信息整合和服务协同是泛在服务的核心。

(7)对称性、融合性 用户不是被动地接受服务,而是可以主动地创造服务;网络作为基础构架,向其他行业提供信息通信服务,实现对信息的综合利用,提升个人、企业、家庭的生活品质及工作效率。

从以上特征可以看出,AUN不是颠覆性的网络革命,而是对传统网络潜力的挖掘和网络效能的提升。

泛在网与物联网、传感器网的关系

泛在网 是指基于个人和社会的需求,实现人与人、人与物、物与物之间按需进行的信息获取、传递、存储、认知、决策、使用等服务,网络具有超强的环境感知、内容感知及智能性,为个人和社会提供泛在的、无所不含的信息服务和应用。

物联网(Internet of things) 是指在物理世界的实体中部署具有一定感知能力、计算能力或执行能力的各种信息传感设备(如传感器、RFID、二维码、短距离无线通信技术、移动通信模块等),通过网络设施实现信息传输、协同和处理,从而实现广域或大范围的人与物、物与物之间信息交换所需的互联。

传感器网(sensor network) 是利用各种传感器(收集光、电、温度、湿度、压力等信息)加上中低速的近距离无线通信技术构成的独立网络,是由多个具有有线/无线通信与计算能力的低功耗、小体积的微小传感器节点构成的网络系统,它一般提供局域或小范围物与物之间的信息交换。

泛在网、物联网、传感器网的关系可以理解为:泛在网是信息社会社会发展的最高目标,物联网是

泛在网的初级和必然发展阶段,传感器网是物联网的延伸和应用的基础。三者之间的关系如图 1 所示。

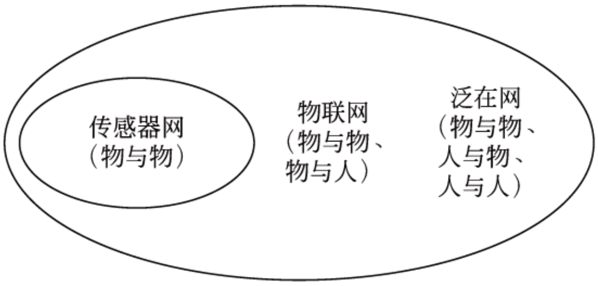


图 1 泛在网与传感网、物联网的范畴

泛在网的整体架构

目前国际上还没有一种为业界所共识的统一的泛在网架构。在国内,泛在网的架构通常被认为是由感知延伸层、网络层和应用层组成的三层结构。

泛在网的感知延伸层包括个域网、家庭网、车域网、AdHoc 网等。个域网、家庭网、车域网是用户服务的网络环境,泛在网络要实现网络无处不在、服务无处不在,需要通信网络来支持网络环境及相关的应用,而 AdHoc 网络是为某些特殊场景或行业的用途而形成的系统或网络,随着无线技术和近距离通信技术的发展,AdHoc 将推动特定场景下的应用,例如军事 AdHoc 网、交通事故下汽车形成的 AdHoc 网,或行业专用的网络,AdHoc 网不仅可以独立工作,还可以和 Internet 或蜂窝无线网络等公众网络进行连接而共同向用户提供服务。感知延伸层面需要增强和优化的特殊要求主要体现在终端设备上,如低功耗需求、低成本需求和防盗防破坏等。

泛在网的网络层可进一步划分为接入网子层、核心网子层及业务能力子层。

接入网子层包括各种有线宽带接入、无线宽带接入、2G/3G/4G 移动接入等。泛在网各类终端和末梢网络的接入会对现有的移动接入产生一定的影响和优化需求。由于泛在网应用有一些新的特征,如 3GPP22.369 中就针对机器类通信归纳给出了 14 个应用特征,包括低移动性、少量数据传送、网络发起的业务、群组管理等。

核心网子层包括现有的电信网、互联网、广电网和各种行业专用网络。目前这些网络基础设施还处于分离的状态,随着网络和业务的发展,这些基础设施将向着融合的方向发展。泛在网业务应用的发展,对现有的核心网还提出了一些特殊要求,如物联网的高带宽需求,大量高突发的小批量数据通信需求以及对各种虚拟专网的支持能力和网络资源的动态调度能力等。

业务能力子层指能够被上层的应用所重复使用的各种业务。现有电信网中的很多业务应用,可以通过开放接口提供给其他应用开发者调用,如会话能力、呈现短消息/多媒体消息等。

泛在网的应用层可以进一步分成应用服务子层和应用支撑子层。应用服务子层针对泛在网所支持的各种业务应用,根据服务对象的不同,可以将泛在网的业务应用分成四大类,即行业专用服务、行业公众服务、面向大众的服务、多行业融合服务。

泛在网并不是要完全重新构建一个全新的网络,而更强调各种网络能力和资源的协同与共享,特别是在现有网络基础上,根据人类生活和社会发展的需求,增加和拓展相应的网络能力、服务和新的应用。

参考文献

1. http://book.51cto.com/art/201006/204047.htm

2. http://news.cnfol.com/110104/101,1587,9096078,00.shtml

(徐明伟)

fanchoulun

范畴论(category theory) 以抽象数学结构(称为对象)和保结构映射(称为态射)为主要研究对象的代数学分支。

范畴的概念于 1945 年出现在 S. Eilenberg 和 S. Maclane 关于同调代数的工作中。现在范畴的语言和基础部分已渗透到数学的很多领域中,并在它们的一些新的发展中起了重要作用。自从 20 世纪 70 年代 ADJ 小组(J. Goguen, J. W. Thatcher, E. G. Wagner 和 J. B. Wright)探讨计算机科学与范畴论的相关性开始,范畴论的一些成果和方法便逐步应用到计算机科学的许多方面,特别是计算机语言学、代数语义学、类型论、形式化技术和软件开发方法等方面。这些应用促进了范畴论的发展。可以相信,像集合论一样,范畴论最终也将找到通向初等水平数学的道路。

范畴 一个"范畴"C 由如下 2 个数学部件组成:

1. 对象(object)簇 $Q(C)$,其元素称为对象,常用 $a,b,c,\cdots$表示;

2. 态射(morphism)簇 $M(C)$,其元素称为态射或箭头。设 f 是 $M(C)$ 中的一个态射,$Q(C)$ 中存在

有序对(a,b)，其中 a 称为态射 f 的定义域(domain)，b 为态射 f 的协域(codomain)。f 称为"从 a 至 b 的态射"，标记为$f:a\to b$。所有从 a 至 b 的态射所组成的集合称为态射集，标记为 $C(a,b)$。所有态射集的并构成态射簇 $M(C)$。$M(C)$ 中态射满足下列 3 个特性：

2.1 态射集不相交性 若 $Q(C)$ 中的对象 a，b，c 和 d 使得 $a\neq c$ 或者 $b\neq d$ 则 $C(a,b)\cap C(c,d)=\{\}$。

2.2 态射组合结合律：二元运算 · 称为态射组合，使得对任意三个对象 a、b 及 c，两个态射$f:a\to b$ 及 $g:b\to c$ 的组合得到一新的态射 $g\circ f:a\to c$。态射组合满足结合律：若$f:a\to b$，$g:b\to c$ 及 $h:c\to d$，则

$$H\circ(g\circ f)=(h\circ g)\circ f:a\to d;$$

2.3 同域态射存在律：对 $Q(C)$ 中任意对象 x，在 $C(x,x)$ 中总存在一个同域态射 I_x：$x\to x$，使得对任意态射 $f:x\to b$，$g:a\to x$ 都会有

$$I_x\circ f=f,\quad g\circ I_x=g$$

如果以集合为对象，集合间的全函数为态射，则可构成一个范畴 SET，称为集合范畴。如果以群、环或域为对象，相应的同态为态射，则可分别构成群范畴 GRP、环范畴 MG 或域范畴 FIELD。如果以半序集为对象，以单调函数为态射，则又可构成半序集范畴 POSET。

如果把函数式程序设计语言 FP 的类型和函数符号分别作为对象和态射，则可把 FP 看作一个范畴。如果把一个逻辑形式系统 FSP 的合式公式和形式证明分别作为对象和态射，则 FSP 也可看作一个范畴。

如果范畴 C 中的全体对象簇和全体态射族均是一集合，则称 C 是一个小范畴(small category)。如果范畴 C 中只有一个对象和一个态射(即该对象的恒等态射)，则称 C 为 1 范畴。

如果把范畴 C 的每个态射 f 都反向，即把 f 的论域和余论域对调，则可获得一个新范畴，称为 C 的对偶范畴，记为 C^{op}，其态射用f^{op}表示。

对偶原则 范畴论中，每一范畴的命题或定理，在其对偶范畴中也成立。这一对偶性在范畴论的任何层次都是普适的。

范畴 C 的图和交换图 构建一个有向图，以范畴 C 的所有对象和态射分别作为该图的顶点和有向边，每一有向边均用对应的态射标识，则称该图为范畴 C 的图表示，简称为 C 的图。如果图中所有有相同起点和终点的有向路径通过组合得到相同的结果，则称该图为范畴 C 的交换图。如果图中每一多边型子图均是可交换的，则该图为交换图。

如果图 1 中的态射 x,y,w,z 满足条件：$y\circ w=z\circ x$，则该图为交换图。如果图 2 是可交换的，意味着图中三态射满足条件：$f=\tilde{f}\circ\pi$。

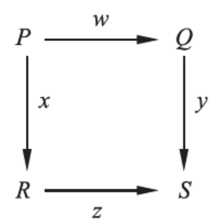


图 1

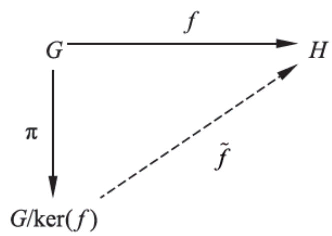


图 2

函子 一个从范畴 C 到范畴 D 的函子被定义为：对 C 中任意对象 X，都有一个 D 中相应的对象 $F(X)$ 与其对应；对 C 中任意态射 $f:X\to Y$，都有一个 D 中相应的态射 $F(f):F(X)\to F(Y)$ 与其对应；并使下列性质成立：

- 对 C 中任意的对象 X，都有 $F(\mathrm{id}_X)=\mathrm{id}_{F(X)}$。
- 对 C 中任意两个态射 $f:X\to Y$ 和 $g:Y\to Z$，都有 $F(g\circ f)=F(g)\circ F(f)$。

函子的实例：给定集合 S，以 S 中的所有元素为对象簇可构成一集合范畴；将 S 中的所有元素存放在线性表 List(S) 中，可以构建以表中的所有元素为对象簇的表范畴。在集合 S 和表 List(S) 之间存在映射 List，它满足范畴间函子的所有特性，所以映射 List 也是集合范畴和表范畴间的函子。

自然变换 一个"自然变换"是两个函子之间的一个关系。函子通常用来描述"自然构造"，而自然变换则用来描述两个构造之间的"自然同态"。有时候，两个截然不同的构造具有"相同的"结果；这正可以用两个函子之间的自然关系来表述。定义：如果 F 和 G 是从范畴 C 到范畴 D 的两个函子，则从 F 到 G 的一个自然变换 $\eta:F\to G$，对于 C 中的

任何对象 X,都有一个 D 中相应的态射 $\eta_X: F(X) \to G(X)$,使得对 C 中的任何态射 $f: X \to Y$,都有 $\eta_Y \circ F(f) = G(f) \circ \eta_X$;这也就是说图 3 是可交换的:

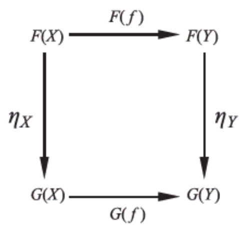


图 3

如果对所有 C 中的对象 X,η_X是一个同构,则从 F 到 G 的自然变换 η 称为自然同构。

泛结构和泛性质 范畴论中定义了初始对象与终止对象,“积”与“和”、外推和回拉、极限和余极限等泛结构。每一种泛结构都是一类满足泛性质的结构的抽象。典型的泛性质是态射的存在性和唯一性。泛结构和泛性质可以屏蔽许多结构重要特性的细节。泛结构和泛性质的应用既可以简化程序正确性证明过程,也可以降低软件和软件开发的复杂性。

目前范畴论的研究成果在计算机科学和软件工程领域具有广泛的应用。

抽象数据类型的求精 如果以抽象数据类型为对象,泛代数中定义的同态作为抽象数据类型间的态射,则可构成一个范畴 ADT,称为抽象数据类型范畴。如果范畴 ADT 中只有一个对象和一个恒等态射,则称该范畴为 ADT1 范畴。在计算机软件方法学中,可以将抽象数据类型的抽象表示和具体表示间的求精或转换关系抽象成函子。可以根据函子的数学特性实现抽象数据类型的求精或转换,以保证其正确性。比如可以将抽象数据类型堆栈和它的数组表示定义为抽象栈范畴和具体栈范畴。可以将抽象栈的求精定义为抽象栈和具体栈之间的同态映射。可以采用两种不同的同态映射得到抽象栈范畴和具体栈范畴之间的两个函子,进而可以通过构造两函子之间的自然变换,得到抽象栈和具体栈等价的结论。

参考文献 1 给出了应用范畴论实现规约合成、证明和构造并发分布式程序的成功案例。

参考文献

1. Fiadeiro J L. Categories for the software engineering. Springer, 2004

2. Pierce B C. Category theory for computer scientists. Cambridge, MA: The MIT Press, 1991

(薛锦云 王兵山 陈意云)

fanshi

范式(normal form) 符合某种级别要求的关系模式的集合(参见**数据库模式**)。国际数据库界制定了一系列构建数据库建议遵循的特殊规则,以确保数据库的**规范化**。在关系数据库里,这种规则就是范式。

数据库设计过程的目标是生成一个关系模式的集合,用这些关系模式来保存数据可以避免不必要的冗余信息。一种普遍采用的判断方法是基于范式的方法。如果一个关系模式满足某种范式,则我们可以知道某些问题可以被避免以及是否需要将该关系模式进一步分解。

根据条件的强弱程度,分别称满足这些条件的关系模式为第一范式(1NF)、第二范式(2NF)、第三范式(3NF)、BC 范式(BCNF)、第四范式(4NF)等。第一范式规定数据库中用以表示实体的属性值必须为不可再分的单个数据。对于第一范式关系:①若非键码属性不函数依赖于任一键的真子集时称此模式为第二范式(参见**数据依赖**、**键码**);②若非键码属性不函数依赖于任一非键码属性时称此模式为第三范式;③所有非平凡函数依赖都是对键码的依赖时称为 BC 范式;④所有非平凡函数依赖和多值依赖都是对键码的依赖时称为第四范式。所有关系数据库实际上都至少是第一范式关系数据库。

第一范式的目的在于简化关系模型,避免复杂的数据结构。其他各级范式都逐级地加强条件使关系模式表达的概念单一化,由此消除更多的数据冗余,从而逐级地加强数据一致性。

参考文献

1. Ullman J D. Principles of database systems. 2nd ed. London: Pitman Publishing Ltd., 1987

2. 施伯乐,丁宝康,汪卫. 数据库系统导论. 3 版. 北京:高等教育出版社,2010

(王鹏 施伯乐)

fanghuoqiang

防火墙(firewall) 在网络边界使用的一种特殊的访问控制设施,它可按预定义的要求限制数据的通过。防火墙通常在内、外网络边界上提供过滤封锁机制。内部网络被认为是安全和可信赖的,而外部网络(通常是 **Internet**)被认为是不安全和不可信赖的。防火墙的作用是防止不希望的、未经授权的通信进出被保护的内部网络,通过边界控制来强化内部网络的安全政策。

防火墙技术主要分为网际协议(IP)报文过滤和应用层代理两类,它们均依据本地的网络安全策略(可表现为防火墙过滤规则),对经过的网络流量进行过滤检查,决定是否允许其通过。报文过滤防火墙又分为无状态检测和有状态检测两种。前者对经过的IP报文进行逐个检查过滤;而后者为TCP连接建立状态,以便为相关的IP报文提供更为精准的过滤能力,例如用于抵御扫描攻击和服务失效攻击。应用层代理防火墙基于代理服务器的功能,对用户的应用访问请求进行逐个过滤并对允许的访问进行桥接。

报文过滤防火墙的常见实现形式是路由器和交换机端口中的访问控制列表(ACL)功能。报文过滤规则通过配置操作定义在ACL中,端口据此对每一个到来的IP报文进行检查,并依据ACL的内容来判断是否对之进行转发。如果是进行有状态检测,则还要看IP报文中包含的高层协议报头的内容(例如TCP连接标志),甚至用户数据部分的内容。如果需要更为丰富的安全功能(例如增加身份认证和安全审计),报文过滤防火墙也可实现成为一个独立的设备,串接在路由器或交换机之前。运行于个人电脑中的个人防火墙(Personal firewall)也是一种报文过滤防火墙。

应用层代理防火墙本质上是应用层网关。用户就一项基于TCP/IP的应用(比如远程通信(Telnet)或者文件传送(FTP))同应用层代理防火墙打交道。防火墙要求用户提供其要访问的远程主机名。当用户答复并提供了正确的用户身份及认证信息后,代理服务器就连通远程主机,为两个通信点充当中继。用户提供的用户身份及认证信息可用于用户级的认证,之后的整个过程可以对用户完全透明。

报文过滤防火墙的优点是结构简单,成本低,且对上层协议和应用透明,因此通用性强。报文过滤防火墙的缺点包括过滤规则的定义与维护的管理负担大,存在被拦截的流量可通过HTTP协议或DNS协议进行封装逃逸问题以及报文过滤的灵活性不够等。

应用层代理防火墙的优点是具有用户级的身份认证、日志记录和账号管理的能力,从而方便网络安全管理员对网络存取和访问进行监控审计。其缺点在于要想提供全面的安全保证,就要对每一项服务都建立对应的应用层**网关**,这就严重地限制了新应用的采纳。

整体看来,由于要求遵从同一组安全规则,防火墙可视作是一种统一的保护功能,因此它更适用于用户具有共同利益的网络环境,例如一个企业网,而不适合像住宅区宽带网这样的网络环境。另外防火墙引入的过滤功能会给网络性能带来负面影响,而且经常会由于过滤规则的不合理而给端系统带来意想不到的影响,并成为网络传输通道的性能瓶颈与单一故障点。需要强调的是,防火墙只能在网络的边界发挥作用,但它不能替代网络内部的安全措施,即它只能解决网络安全的部分问题。

参考文献

Cheswick WR, Bellovin SM, Rubin AD. Firewalls and Internet security: repelling the wily hacker. 2nd ed. Boston, MA: Addison-Wesley, 2003

(胡道元　龚俭)

fang xinxi xielou jishu

防信息泄露技术(technique of electro-mechanical protection against encission and spurious transmission, TEMPEST)　研究抑制计算机及其外围设备等信息设备的电、磁、声等信号杂散发射的技术。

计算机等电子设备工作时,会向空间发射电、磁、声信号,这些信号一旦被他人接收分析,即可得知计算机所处理的信息内容,从而造成泄密。随着科学技术的发展,有时使用不太复杂的设备也可截获计算机的各种信息。例如,可在1 km以外用一种不太复杂的装置接收并复现计算机显示器屏幕的信息。计算机的信息泄露不只限于显示屏幕,计算机主机、**键盘**、打印机等都会造成信息泄露,只是其接收难易程度不同而已。

TEMPEST一词最初是美国政府一项绝密计划即控制电子设备泄密发射的代号。后来成了研究各种信息设备泄密发射的代名词。防信息泄露技术研究主要包括以下三方面的内容。

(1)标准及规范研究　标准及规范研究的主要内容是信息泄露极限值、信号检测方法、设备使用环境要求以及使用环境的评估。美国在20世纪70年代已制定了一套较完善的TEMPEST标准和规范(即NACSI M5100系列标准)。随着技术的进步,该系列标准分别在1974年和1984年进行了更新。90年代初,随着冷战的结束和国际形势的变化以及TEMPEST技术发展的实际情况,美国对原有的TEMPEST标准作了重新修改并再度颁布,新的标准

包括 NSTISSAM、NSTISSP、NSTISSI 等系列标准。新标准的最大变化在于给防护设备分级，对使用环境分级，以适应不同用户的需求。北约在 1982 年颁布了自己的 TEMPEST 标准，即 AMSG 系列标准，分别为 AMSG 720B、AMSG 788、AMSG 784。英国的 TEMPEST 标准为 BTR/01/202。我国于 1999 年颁布了自己的 TEMPEST 标准。

（2）防护及制造技术研究　虽然屏蔽室、干扰器和 TEMPEST 都可以起到防信息泄露的目的，但只有 TEMPEST 是防止计算机信息泄露的根本措施。防信息泄露设计中的一项核心工作是红信号和黑信号的划分，包含机密信息且又未经加密的信号称为红信号，存在红信号的区域称红区。不包含机密信息或经加密处理过的信号称为黑信号，只存在黑信号的区域称黑区。防护信息泄露的基本措施是抑制红信号的发射和防止红信号在红区和黑区之间传输。为了抑制红信号的发射和传输，常采用两种方法：一种是包容法，主要采用屏蔽、滤波、隔离等技术对设备内部各单元或设备本身进行设计或改造，从而抑制红信号的传输及发射。另一种方法是抑源法，主要是从产品最初设计阶段开始，从电路设计、布线及元器件选择等方面抑制红信号的传输和发射，从而使最终产品满足防信息泄露要求。传统的防护技术必须采取一定的物理处理措施，设备的改造成本较高。后来有人提出 Soft-Tempest 技术概念，其技术优点在于利用软件来降低视频信号的辐射发射，防护成本低。

（3）检测技术研究　主要包括红信号检测方法、测试系统组成与专用检测设备的研制等内容。

红信号的检测分析是防信息泄露技术特有的一项内容。常采用特征识别法和相关比较法，通过测量信号脉冲宽度、波形上升及下降时间、脉冲重复周期等参数，从时域和频域分析信号特征，从而最终区分红黑信号。

防信息泄露检测所需设备主要包括 TEMPEST 测试接收机，各种天线及探头，信号分析仪器及屏蔽室。其中 TEMPEST 测试接收机为防信息泄露测试专用设备，要求设备有较高的灵敏度，中频带宽较宽且可调节，中频波形因数好，自动化程度高，并具有高性能的前置及后置滤波器以及多种解调及输出方式等。

（於亮）

fangsheng jiqiren

仿生机器人（biologically inspired robots）　基于自然生物的结构、形态、运动、内在机制和环境适应性，设计开发的可以模仿类似自然生物的功能、结构、内在机制的机器人。仿生机器人通常也被称为受生物学启发的机器人（bio-inspired robot），或模仿生物的机器人（bio-mimetic robot）。受生物学启发的机器人研究倾向于将对自然生物观察中抽象的某些原则改造成为工程方法应用于机器人，而模仿生物的机器人研究则倾向于通过尽可能复现对生物观察中获得的机理和过程来代替传统的工程解决方案在机器人中应用。

仿生机器人涉及机器人的机构、运动学和动力学分析、驱动系统、感知系统、控制系统、智能决策系统、信息传输机制、能源供给系统、人机接口等很多方面的仿生机制。

形态与运动　自然界生物通过进化形成了与其生存环境高度适应的形态和运动方式，因此，可以通过研究这些生物的自然形态和运动方式来设计出高效率、高机动性、具有环境适应能力的机器人系统。模仿生物行走的机器人，如双足仿人机器人、四足仿生机器人（机器狗、机器猫、机器骡马等）、六足仿昆虫机器人（机器蟑螂、机器蟋蟀等）、仿蛇型机器人；模仿生物攀爬的机器人，如仿壁虎机器人、仿长臂猿机器人；模仿生物游动的机器人，如仿鱼机器人、仿水母机器人、仿多毛类蠕虫机器人；模仿两栖生物运动的机器人，如机器龙虾、机器螃蟹、机器海龟；模仿生物飞行的机器人，如扑翼飞行机器人（仿海鸥机器人、仿蜂鸟机器人、机器飞蛾、机器苍蝇等）。如图 1 所示，几类不同形态的仿生机器人，四足机器人、机器鱼、机器蜻蜓、机器龙虾、机器壁虎、机器蛇等。这些机器人系统独特的仿生运动方式和形态可使其在特定环境中显示出良好的机动性、隐蔽性和适应能力。

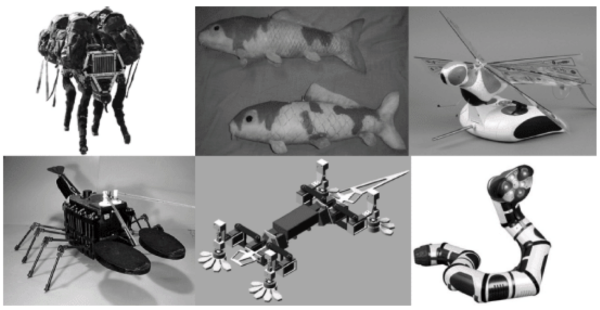


图 1　几类不同形态的机器人

感知与处理　自然生物的感知系统复杂而精妙，具有极高的灵敏度、抗干扰性和准确性，因此，可

通过对自然生物感知系统的研究来形成机器人视、听、触、味、嗅、体感等信息采集与处理的系统和方法。在视觉方面,通过模仿生物的趋光性解决一些避障、通信方面的问题;借鉴苍蝇复眼的结构设计机器人视觉光电系统;仿照蚂蚁定位机制设计光流法和里程计法用于机器人导航;借鉴人的视觉系统构建机器人的环境识别、定位、地图构建方法。在听觉方面,模仿蟋蟀的听觉结构设计机器人的声源定向系统;研究蝙蝠的声音认知能力,发展机器人的声信号获取与处理系统;模仿蝙蝠或海豚的声呐机制建立机器人回声定位系统。在触觉方面,模仿蟑螂的触须设计机器人的避碰控制系统;研究老鼠通过胡须感知环境的机制设计机器人的触觉传感器及其信号处理系统;也有利用半导体触觉传感器的压力信号或薄膜式触觉传感器的光电信号来模仿人的触觉。

控制与智能 自然生物为了生存而发展出复杂的内在控制机制和极强的环境适应能力,因此,可通过自然生物内在控制机制和认知机制的研究开发出更有效的机器人控制体系结构和方法。在机器人实时控制方面,基于中枢模式发生器(CPG)等神经网络模型的控制,模仿人的控制行为的模糊控制等都有较多的研究和应用。在机器人的控制体系结构方面,借鉴生物学的研究工作,形成了许多不同类型的机器人体系结构。在受神经科学等启发而发展出的基于行为的控制体系结构基础上已构建了很多具有最小认知结构的机器人系统。基于大脑神经组织结构分析与假设而形成的基于学习的控制体系结构,如仿生联想式学习、增强式学习、模仿式学习等,也在一些机器人系统中应用验证。此外,基于生物进化机制的机器人系统自主优化及进化的方法,模仿幼儿认知过程的成长型机器人的学习方法也都有一些研究和应用。

仿生的手段,不仅在结构、感知和控制方面可以为机器人研究提供新的思路,在能源供应、人机接口等方面也具有良好的应用远景。如模仿动物的捕食和消化过程设计新型的机器人能量供应系统;模仿生物群体行为的协作作业系统;采集、分析并识别脑电信号、肌电信号、神经信号等,并用于机器臂、智能假肢的控制。此外,利用活体神经细胞或细胞群接收机器人传感器信号并控制机器人执行机构运动也可视为仿生机器人的研究范畴。

参考文献

1. Pfeifer R, Lungarella M, Iida F. Self-organization, embodiment, and biologically inspired robotics. Science, 2007, 318: 1088-1093

2. Siciliano B, Khatib O (eds.). Handbook of robotics. Berlin: Springer-Verlag, 2008

3. Vepa R. Biomimetic robotics: Mechanisms and control. Cambridge University Press, 2009

(王硕　谭民)

fangzhen xunlian xitong

仿真训练系统(simulation training system)

通过综合运用**计算机仿真技术**、**虚拟现实**技术、**人机交互技术**、自动化技术等手段,将计算机系统、物理系统、训练者、训练环境等集成起来,以用于人员训练的一种专用设备。它一般由以下4个部分组成:

(1) 计算机系统　它完成模型运算、数据转换、通信以及各种数值计算或符号处理的工作。

(2) 训练环境　它为被训练者提供一个与真实系统相类似的环境,如飞机座舱、电站控制室等。

(3) 教员系统　它使教员能设置训练开始时的初始条件,给定训练科目,并记录训练结果,显示学员的训练过程,从而可对训练成绩进行评定。

(4) 人体感受系统　它将计算机输入和输出的数字数据,都转换为人体的真实感觉,作用于被训练者。比较典型的是视觉(图像仿真技术)、听觉(声响仿真技术)和身体姿态、加速度(运动仿真技术),以及触觉和操作力度感觉等。

凡是需要一个或一组熟练人员进行操纵、控制、管理和决策的真实系统,都需要对人员进行训练。对于复杂的系统,采用仿真训练系统进行训练不仅能降低成本、提高安全性,而且能方便地安排训练计划,提高训练的效率和质量。有些训练(如危险事故处理)不可能在真实系统上进行而却能在仿真训练系统上多次重复地进行,因此仿真训练系统具有极高的经济效益与社会效益。

20世纪40年代,美国为训练空军飞行员首先研制成功了飞行训练器,当时的模型计算装置是机电的,训练的科目也极为有限。经过几十年的发展,现代仿真训练系统不论在技术水平上,还是在应用领域上都取得了长足的发展。在工业领域中,核、火、水电站、变电站、电网调度、石油、化工厂等连续生产部门,相继开发出操作人员培训仿真系统。交通部门则已拥有了飞机、船舶、火车/地铁和汽车驾驶培训仿真器以及航空港指挥调度仿真系统,城市

交通管理仿真中心,集装箱码头的岸吊仿真器等。在医学应用方面,已经开发出外科和牙科手术实习培训和预演仿真系统。建筑行业中已有塔吊司机培训仿真器。在军用领域,如海陆空三军指战员的各类武器、装备操纵和战术训练系统等,更是发展最快的领域,仿真训练系统已成为不可或缺的装备之一,以致著名的英国简氏武器年鉴专门编辑出版了军训仿真器期刊。此外,在娱乐业中,不断出现如动感电影院、穿梭机、多维仿真馆和各种新兴的娱乐仿真、虚拟现实设施等。

仿真训练系统已应用到很多领域,从仿真训练目标的角度可分为以下 3 类:

(1) 载体操纵型　如飞机起降仿真训练系统、舰船操纵仿真训练系统、车辆驾驶仿真训练系统等。它以训练操纵这些运载工具的人员为主要目的。

(2) 过程控制型　如电站仿真训练系统、化工系统仿真训练系统等。它以训练控制这类复杂系统的主控人员为主要目的。

(3) 博弈决策型　如飞机格斗仿真训练系统、战术战役仿真训练系统、调度管理仿真训练系统等。它以训练军事指挥员、经理调度员以及其他各类决策人员的决策能力为主要目的。

仿真训练系统的共同关键技术是:①研究与开发适合于训练要求的数学模型,并建立一套先进的建模环境,以便缩短训练仿真器的研制周期;②各种环境仿真的技术,如视景、音响仿真、具有加速度、不平衡运动、失重状态、运动感觉、力感觉的仿真等;③多功能、智能化教员系统和人体感受系统。

仿真训练系统中广泛应用了计算机仿真技术,特别是实时仿真技术,比如,利用计算机图形与图像技术产生真实环境的景像;利用**数据库**管理各种模型并建立相应的模型库;利用**人工智能**技术对训练结果进行评价等。近年来,虚拟现实技术、计算机仿真技术的发展更加提高了训练仿真器的逼真度,它使被训练者与由计算机产生的"虚拟环境"更加紧密地融合在一起,使被训练者有一种完全身临其境的真实感。

参考文献

1. 王扬. 现代训练仿真器技术的发展. 计算机仿真. 2003, (1)

2. 吕崇德,徐忠净. 训练仿真器. 中国计算机用户,1990,6

(王威　肖田元)

fangzhen yuyan

仿真语言(simulation language)　专门用于仿真的面向问题的非顺序性的计算机语言。仿真语言为仿真人员提供仿真建模、仿真实验和仿真结果分析、显示等能力(参见**计算机仿真**),使得仿真活动可以不依赖于掌握通用计算机高级语言的编程细节和技巧,从而把主要精力集中在仿真研究上。

早期的仿真都是仿真人员根据模型及仿真要求采用通用程序设计语言来设计仿真程序,这就要求仿真人员不但要熟悉仿真的对象,而且要熟悉计算机及**程序设计语言**,大大限制了计算机技术在仿真中的应用。

为简化仿真模型的准备及编程工作,各种通用的仿真程序包相继产生,它实际上是一种根据某类系统仿真需要而事先编好的程序,仿真人员只需按照其事先设计好的格式来输入有关参数、数据,仿真程序包就可自动构造出仿真模型,编排好仿真计算次序,并可加以执行,输出仿真结果。这样研究人员就可将精力集中在所研究的对象上,而不必花费大量时间用于程序的开发与调试。这对于推广计算机在仿真领域的应用起了极大的促进作用。

仿真程序包是仿真语言的初级形式。尽管仿真程序包具有一定的通用性,但由于是事先设计好的程序,因而模型描述能力、模型执行控制能力及输入方式及输出分析能力总是有限的。为适应更为复杂的系统仿真建模的要求以及满足各种仿真运行的需要,在仿真程序包的基础上产生了仿真语言。

仿真语言与仿真程序包的最大区别在于,仿真语言有一套完整的描述仿真对象及仿真运行所需要的符号、语句及语法,能检测出语法错误及逻辑错误。

仿真语言的发展历程大体上可分为四个阶段:

第一阶段为初级语言阶段(1960—1970 年)。代表性的有:面向连续系统(参见**连续系统仿真**)的 MIMIC, MIDAS, DSL/90, CSMP, CSSL, DARE 等;面向离散事件系统(参见**离散事件系统仿真**)的 GPSS,GASP 等,它们提供统一的建模、实验、统计输出的框架。

第二阶段为高级仿真语言阶段(1970—1980 年)。代表性的有:面向连续系统的 CSSL IV, DARE-P, ACSL 等;面向离散事件系统的 GPSSIV, SIMSCRIPT Ⅱ.5, SLAM, SIMULA 等。其特点是具有多种建模观点,仿真过程具有跟踪能力,增加了统计分析功能;特别是,由于计算机技术的发展,仿真语言的宿主机逐步过渡到以小型机为主。

第三阶段为平台仿真语言阶段(1980—1990

年)。代表性的有:面向连续系统的 TESS、GEST、MATLAB 等;面向离散事件系统的 SLAM Ⅱ, SIMAN 等。其特点是具有在数据库基础上实现数据存储与检索、脚本仿真、数据收集、数据分析、报告及图形生成、脚本动画、网络模型输入、运行控制、数据管理等功能的集成,是一体化仿真环境。

第四阶段为多领域仿真语言阶段(1990—2000年)。代表性的有:面向连续系统的 MATLAB 7.0 版,OpenSML,MODLICA 等,面向离散事件系统的 AutoMod, Extend, Flexsim, Micro Saint, ProModel, QUEST,SIMUL8,WITNESS 等;还有一些如 Arena, AnyLogic 既可用于离散事件系统、连续系统,也可用于连续-离散事件混合系统建模与仿真。其特点是提供完整的工程设计仿真支撑环境,提供应用库,库中来自不同物理领域的模型都是经过严格的测试和实验验证的,用户在该环境下可以建立复杂的多学科领域系统的模型,并在此基础上进行仿真计算和深入的分析,使得用户从烦琐的数学建模中解放出来从而专注于物理系统本身的设计。

仿真语言评述

国内外流行的仿真语言有许多种,它们分别应用于不同的领域。按照系统模型的种类,可分为两大类,一类称为**连续系统仿真**语言(面向常微分方程、偏微分方程、差分方程模型),另一类称为**离散事件系统仿真**语言(面向离散事件系统模型,如排队模型、PERT 网模型、Petri Net 模型等)。也有一些语言同时具有上述两种模型仿真的能力,但其实质仍然是为了解决两类模型在仿真时的相互控制和通信技术,从仿真建模方法学的观点来看仍然可按两大类考虑。

对目前流行的仿真语言从 4 个方面加以评述。

(1) 表达模型的能力　许多先进的连续系统仿真语言,均可面向问题方式,用多种形式(文字、方程、图形)表示常微分方程描述的模型(包括方程组、向量、间断右端函数、传递函数、框图等形式)和差分方程描述的模型(包括方程型及 Z 变换形式)。至于对偏微分方程(包括抛物线型、椭圆型、双曲型、双调谐型)模型的仿真,则趋向于做成专门求解某一类型的专用程序包形式。

离散事件系统仿真语言一般采用事件调度、活动扫描、进程交互、三阶段(three phases)中的一种或多种策略来描述模型。此外,人们还结合具体应用领域(如物流系统、制造与物料储运系统、计算机系统、网络系统等)发展更为专用的面向问题描述的离散事件系统仿真语言。

(2) 语言结构特性　语言的结构特性包括既要便于建立模型和模型执行,又需考虑其翻译的可能性。

在连续系统仿真语言中,大多允许以图形方式按任意顺序描述模型,即具有所谓"并行"结构,仿真语言具有排序算法。图形工具以结构图为主,每种语言都具有自身的图元库,用户可交互式从图元库中选择模型、定义参数,建立图元之间的连接关系,并具有灵活的编辑功能。许多连续系统仿真语言中引入了"子模型"的概念,有些还实现了"段"结构,以适应多帧速、实时仿真及多处理机仿真的需要。

离散事件系统仿真语言常借助于流程图、活动周期图、PERT 网等图论工具进行描述,实现并行结构,并以过程并发的形式实现仿真钟的推进。

结构化递阶建模的思想已在新的语言中越来越多地得到使用,有些语言还实现了模型与实验框架分离描述,从而实现模型的数据驱动。

(3) 执行方式　早期的仿真语言大多是非交互式的。近年来,交互式语言已成为发展方向。大多数流行的仿真语言可以交互改变模型参数、初始条件、控制仿真实验过程等(无须重新编译、连接),但是模型结构的交互改变还只见于少数的仿真语言中。

(4) 输入输出能力　早期的仿真语言其输入一般采用文件方式,而其输出是标准的数据报告。目前,"视算"的思想已广泛应用于仿真语言中,用图形、选单、表格、图符等方式输入模型及参数,以漂亮的图形、动画(实时的、二维的或三维的)输出显示已是普遍的现象。

近年来,国际标准化组织提出的模型驱动体系结构 MDA(model driven architecture)在计算机信息领域引起广泛重视,并将对建模仿真技术的未来发展产生重要影响。MDA 试图采用统一建模语言 UML 等抽象语言对软件的设计模型进行统一描述,并在一系列基础构件和商用技术标准的支持下,自动生成C++代码、Java 代码、C#代码、测试框架、整合代码、Java 对象、.Net对象、部署脚本乃至软件文档,而且可以将其映射到 J2EE、.Net、CORBA 等平台,以最大限度地提升软件的开发效率和可重用程度。借鉴 MDA 思想,可以建立具有较高平台无关性的仿真语言及仿真标准,用以支持仿真模型的设计、存储、交换、执行、可视化等功能的实现,并促进仿真

模型组合、重用等性能的提高。

参考文献

1. 文传源. 系统仿真学科与仿真系统技术. 系统仿真学报,1992,4(3)

2. Jerry B, et al. 离散事件系统仿真. 肖田元,范文慧,译. 北京:机械工业出版社,2007

(范文慧　肖田元)

fei chuantong jisuanji

非传统计算机(non-traditional computer)

不采用冯·诺依曼结构的电子计算机。其工作原理不同于传统的冯·诺依曼计算机,也不同于在冯·诺依曼计算机基础上发展起来的其他计算机。

冯·诺依曼等人于1945—1946年间提出了一种计算机的系统结构,这种系统结构采取了存储程序方式,指令和数据一起存在存储器中,存储器按地址访问,在中央处理器(CPU)的控制器中设有一个指令计数器,指令按指令计数器指示的顺序逐条地串行执行,后人称这种系统结构为冯·诺依曼结构。按这种结构做成的计算机称为冯·诺依曼计算机。

其后,随着计算机器件、硬件和软件的发展,对最初的冯·诺依曼系统结构作了很多改进,改进的主要目的是要增加计算机的并行处理能力,提高处理速度。采取的主要手段是时间上的重叠和空间上的并行,可以是:操作间的重叠(运算流水线)、指令间的重叠(指令流水线)、向量执行的重叠(向量处理机)、指令级并行(多功能部件、超标量计算机、超长指令字计算机)、多进程-多任务、多线程、用许多个CPU并行工作(多处理机)和分布式处理(网络计算、机群)等。经过这些改进后,指令计数器多了,指令也不再是逐条地串行执行了。但是它们都没有改变主存储器按地址访问,指令执行的次序由指令计数器控制的基本原理,这些计算机都称为**控制驱动**的计算机。所谓控制驱动,就是在计算机中,只有当指令计数器指向某条指令时才驱动该条指令的执行。控制驱动的计算机称为传统计算机,即冯·诺依曼计算机。

为了适应对计算机运算速度不断提出的更高要求,从20世纪60—70年代起,有人提出多种摆脱冯·诺依曼计算机原有模式束缚的设想,以求在计算机系统结构方面取得更大的突破,开发出具有更高并行功能和系统效率的非传统计算机。为此进行了一系列科研和实践,到80年代已发展出多种类型的非传统计算机。在这些非传统计算机中,有的摆脱了控制驱动结构的束缚,采取了数据驱动结构(**数据流计算机**采用这种结构)或需求驱动结构(**归约机**采用这种结构);有的突破了存储器按地址访问的框框,采取了按内容访问存储器(数据库机主要采用这种结构);有脱离了以数值处理为主的传统计算机模式,开创以符号处理为主的面向智能知识信息处理的智能计算机,等等。

主要特点

在摆脱控制驱动结构的束缚方面　在实际程序中可能存在着大量能够并行执行的指令,但在传统的冯·诺依曼计算机中,由于受到了控制驱动下的串行执行的限制,难以并行执行这些可以并行执行的指令。在数据流计算机(参见**数据流计算机**)中取消了指令计数器,采取数据驱动方式启动指令的执行。数据驱动的含义是:程序中的任一条指令只要其所需的操作数已经全部齐备,就可以立即启动执行。一条指令的运算结果又流向下一条指令作为下一条指令的操作数来驱动该指令的执行。这样,所有有条件执行的指令都能并行执行,而不必等待指令计数器的驱动(其前提应是有足够多的硬件执行部件等资源)。如能这样,显然其并行处理能力会远高于传统的计算机。

需求驱动的系统结构与数据驱动的系统结构有些相似,但它们执行操作的次序不同。在数据驱动系统结构中,指令在它的全部输入操作数到齐时立即开始启动执行,这样做的结果难免要多执行很多本来不需要执行的指令。在需求驱动系统结构中,程序仅在需要用它的输出结果时才开始启动。如果此时这条指令的输入数据还未获得,则由这条指令再去启动能获得它所需的各个输入数据的指令,这样就可以把需求链一直延伸下去,直到需要的外部输入数据到达为止。当需求的数值都已到齐,需要它的指令才能继续执行下去,这就是需求驱动的含义。采用需求驱动可以使计算机只需要执行最低限度的计算工作,从而提高计算机的工作效率。归约机就是这样一种需求驱动的计算机(参见**归约机**)。

在以符号处理为主的智能计算机方面　与数值处理比较,符号处理的主要特点有:用符号的**知识表示**,密集的搜索操作,庞大的存储容量而且不存在访问局部性,消息长度可变,采用不确定算法,用交互的I/O和对知识库的需要。传统计算机的设计适合于作数值计算,不适应符号处理,在进行符号处理时效率低下。因而人们提出了以符号处理为主的智能计算机的设想。到20世纪80年代已研制的智能计

算机可分为3类：基于语言的智能机、基于知识的智能机、智能化I/O接口机。

（1）基于语言的智能机的设计目标是高效地执行面向人工智能的高级程序设计语言。机器中设有专门的硬件来实现所支持的语言的基本操作，大多是单处理机或只用少数几台高性能处理机组成的粗粒度并行的系统。这类机器又可分为3种：LISP机、Prolog机、函数式语言的智能机。LISP和Prolog是两种最主要的应用于人工智能的高级程序设计语言。

LISP是以表作为处理对象的程序设计语言（参见**LISP语言**）。LISP机是专为支持LISP语言而研制的计算机，它的指令集与LISP语言的基本函数相接近，可以直接执行用LISP语言编写的程序。它对指令集中出现频度高的操作尽量加快执行速度，并采用面向栈的有标志位的数据格式，对LISP语言所具有的动态类型检查、动态嵌套、无用存储空间回收和大量函数调用的特性，在系统结构上都有所考虑，使它在执行LISP程序时达到很高的运行速度。LISP机的研究始于1975年的CONS机，第一台商品化的LAMBDA LISP机出现于20世纪80年代初。80年代在市场上还曾有过多种LISP机产品，较重要的有Symbolics 3600系列、TI Explorer、Xerox 1100系列等。

PROLOG是一种顺序逻辑程序设计语言（参见**PROLOG语言**），PROLOG机是直接执行用PROLOG语言编写的程序的计算机，它以模式搜索和合一为基本机能，以归结原理为基础进行逻辑推理。80年代日本在“第五代计算机系统”计划中曾研制过的**逻辑推理机**即属于PROLOG机。

冯·诺依曼计算机中使用的传统程序设计语言，像FORTRAN，PASCAL等都是命令式语言。由于命令式语言限制了计算机并行性的开发。命令式语言的结构必须服从于串行的冯·诺依曼计算机系统结构，难以全面反映所解题目的本质。函数式语言就是为解决此问题而提出的。**函数式程序设计**语言需要有适合于它的全新的系统结构。归约机就是一种函数式语言计算机。

（2）基于知识的智能机的关键部分是知识的表示与操作。为了模拟人脑对知识的并行处理过程，这类机器中包含数量众多的处理器，每个处理器的硬件都比较简单并有一个小容量的局部存储器，通过大量处理器进行高度并行的工作来实现对知识的智能化处理。基于知识的智能机又可分为：语义网络、基于规则的系统、基于目标的系统和神经网络。

（3）智能化I/O接口机的功能是提供人工智能系统需要的语言识别、**自然语言理解**、**模式识别**、图像处理、计算机视觉等人机间的智能化接口。

在上述这些智能机中，除了有一些LISP机和极少数面向人工智能的机器（如connection machine）有商品化的产品外，其余的都是大学或工业部门的研发实验室中的探索性系统。

在数据库机和知识库机方面 传统计算机不适合于数据库应用中所需的查找、排序、检索、更新、插入、删除以及数据转移等操作，这些操作使数据库管理系统软件的结构复杂，效率不高。20世纪70年代初，随着数据库管理系统（特别是关系型数据库）的推广应用，开始了数据库机的研究。***数据库机***是一种专门用来替代由数据库管理系统软件所实现的大部分或全部功能的计算机或协处理器。在数据库机中，设置了专门的硬件来提高数据库服务的速度，如外存储器采用关联存储方式以打破磁盘输入输出的瓶颈，用硬件直接进行连接和排序等操作以提高处理速度，采用功能分布式系统进行并行处理等。数据库机曾有过多种实现方案，值得注意的是在网络环境中或具有客户-服务器计算模式的分布式处理环境中的数据库服务器，它是专门用来为客户提供数据库服务的独立的计算机（参见**服务器**）。

知识库机是一种专门针对知识库管理和知识处理的特点而设计的计算机。知识库机通常总是和智能计算机联系在一起，往往隐含在非冯·诺依曼计算机中。如果把数据库中储存的内容看作是一类断言性知识，就可以把数据库机作为知识库机的组成部分。

展　望

进入20世纪90年代后，传统计算机的工作频率、存储容量和存取速度、并行处理和网络通信能力都有长足的发展，很多当年想通过突破传统计算机的束缚来达到高性能目标的应用项目已经能在传统计算机的平台上高效率、低成本地完成。高性能的数据库服务器已足以满足共享网络信息资源的需要。因此数据驱动和需求驱动的计算机以及数据库机和知识库机的研制工作基本上都停顿下来。同时，LISP语言本身有了很大的发展，扩充了新的功能，LISP机的研究趋向于在传统的通用计算机平台上改进编译技术，增加专用部件或适当修改某些硬件，以达到高性能、低成本运行LISP程序的目的，已

不再研制专门的 LISP 机。但是非传统计算机的某些研究成果在后来的传统计算机中得到应用。例如,数据驱动原理已用到超标量计算机的乱序发送中;按内容访问的技术已用到高速缓冲存储器中。

在智能机方面,从 90 年代开始,神经网络得到了很大的发展。在人工神经网络的基础上构建了神经处理机(参见**神经计算机**)。人工神经网络已能有效地在投资分析、签名分析、过程控制、飞船和机车发动机监控等项目中得到应用。在语言识别、图像识别、工业机器人、医学图像处理、数据采集等领域,都有卓越的表现。

传统的微电子技术在集成度和工作频率方面已发展到接近于其物理极限。为适应未来继续发展的需要,人们还在研究生物分子器件(参见**生物计算**)、光器件(参见**光计算机**)和量子器件(参见量子计算机),用这些全新的器件有可能组成生物计算机、光计算机和量子计算机。 (孙强南)

feidandiao luoji

非单调逻辑(nonmonotonic logic) 泛指常识推理形式化研究中提出的具有非单调性(增加前提的信息量却有可能减少可推出的结论)的逻辑系统。建立能够模拟人类常识推理的系统是**人工智能**研究的重要目标之一,构建刻画常识推理的形式化逻辑系统是实现该目标的一个关键步骤。经典逻辑旨在形式化数学中的推理,著名哲学家 L. Wittgenstein 早就指出它不适合常识的形式化。20 世纪 70 年代末期,学术界相继提出各种逻辑系统用于形式描述常识推理。因为这些系统的推理表现出非单调性,所以将它们统称为非单调逻辑①。具有代表性的非单调逻辑理论包括:R. Reiter 的**缺省逻辑**、J. McCarthy 的限定论、D. McDermott 与 J. Doyle 的模态非单调系统、R. C. Moore 的自认知逻辑以及由 D. M. Gabbay、D. Makinson、Y. Shoham、S. Kraus、D. Lehmann 及 M. Magidor 等人发展起来的非单调后承理论。

常识推理中出现非单调现象是非常自然的。首先,常识推理往往是在信息不完全的情况下进行的,为了有效地进行推理经常采用如下所谓"jumping to conclusions"的推理模式:

> if there is no evidence that would contradict A,
> then concludes A.

显然,在这种推理模式下,增加新的信息可能会导致原来在该模式下可推出的结论失效,从而出现非单调性。其次,在常识推理中,人们通常不会平等对待所有模型,有些模型是人们所预期的,常识推理期望获得的是在这些预期模型中成立的结论。而增加新的前提条件,会使预期模型发生变化,从而也可能导致非单调性的出现。

非单调逻辑系统一般由两部分组成:系统所基于的单调逻辑以及该逻辑上的非单调语义。其中,单调逻辑提供理论的含义或逻辑内容,而非单调语义将决定该理论在此非单调逻辑下的后承。按照 A. Bochman 的分类方法,非单调逻辑系统大体上可以分为解释型和择优型两类。前者包括缺省逻辑以及模态非单调逻辑等早期出现的系统;后者则以非单调后承和信念变化理论为代表。限定论既可以看成解释型的也可以看成择优型的。解释型和择优型系统有着诸多的不同之处。首先,两类非单调逻辑系统都涉及假设的使用,均可以看成是关于假设的合理使用的理论。假设在两类系统中发挥的作用是不同的。解释型系统将假设当作猜想,用于解释观测;而择优型系统则将假设当作默认的事实。其次,预期模型在两类系统中是按照不同的方式引入的,例如,缺省逻辑中缺省理论的预期模型(即,扩张)是通过不动点方式获得的,而 Kraus,Lehmann 及 Magidor 关于非单调后承的工作则是基于择优序引入预期模型(即,最"正规的"可能世界)。Shoham 曾经希望通过推广择优序将缺省逻辑纳入择优型非单调系统范畴,但后续的研究表明缺省逻辑不满足累积型非单调后承的基本假设,从而导致学术界对此设想的质疑。

与通常的逻辑系统研究类似,对非单调后承研究的一个重要方面是对其语义的研究。该领域主要采用的语义是择优语义,这种语义体现了对可能世界的选择性。在非单调逻辑研究领域,择优语义的思想最早出现在 McCarthy 的限定论中,他采用的选择标准非常具体——对"反常谓词"采用极小化解释。Shoam 进一步将选择机制用抽象的序关系加以刻画,从而形成所谓择优模型。需要指出的是,早在 20 世纪 60 年代,逻辑学家 B. Hansson 研究道义逻辑时已经提出类似的模型。

完备性与可靠性定理是经典逻辑中联系语形和语义的基本元定理,在非单调后承研究中,表示定理扮演着类似的角色。这类定理揭示了择优模型的结

---

① 实际上,具有非单调性的逻辑系统在此之前早就已经存在了,例如,概率逻辑系统。

构性质与非单调逻辑规则之间的联系。非单调后承研究中第一个非平凡语言下的表示定理是由 Kraus, Lehmann 及 Magidor 建立的。该表示定理涉及的后承关系局限于公式与公式之间,对公式集合与公式之间的后承关系相应结论不成立。在经典逻辑中,这两种后承关系无本质区别,但对非单调后承而言,由于单调性与紧致性的缺失,两者有根本性的差异。

20 世纪 80 年代至今,非单调逻辑的理论研究取得了比较丰硕的成果,然而,非单调推理系统的成功应用案例却很稀少。究其原因主要包括两方面。首先,常识推理往往涉及多种属性的推理(例如,时间、空间、意图、目的等),非单调逻辑只有和面向其他属性的推理系统有机结合,才有可能得到应用。但这种结合往往非常困难,例如,动作推理中的 Yale 射击问题表明将时序推理与非单调推理相结合是非常复杂的;非单调推理与多 agent 系统的结合也是困难的,这涉及 agent 对其他 agent 的非单调推理过程进行推理,大多现存的非单调推理系统都不具有表达这种嵌套非单调推理的能力。其次,多数非单调推理系统的推理机制涉及语句集合的协调性判定,这对具有较强表达能力的推理系统的机器实现设置了理论障碍。具体而言,由于一阶理论的协调性是不可判定的,因此基于协调性判定的谓词非单调推理系统是不可机器实现的;虽然在命题逻辑语言下协调性是可判定的,但它属于 NP 完全类,目前普遍认为这类问题没有多项式时间复杂度的算法,所以,基于协调性判定的命题非单调推理系统几乎是不可能高效实现的。当然,通过对语言表达能力进行限制,可以实现具有较高效率的非单调推理系统,例如,采用了"失败即否定"机制的逻辑程序设计语言就可以比较高效地实现一些类型的非单调推理。

参考文献

1. Gabbay Dov M, Hogger C J, Robinson J A. Handbook of logic in artificial intelligence and logic programming, Volume 3. Clarendon Press, 1994

2. Marek V W, Truszczynski M. Nonmonotonic logic. Springer-Verlag, 1993

(朱朝晖)

feiguocheng yuyan

非过程语言(nonprocedural language) 不显式指明处理过程细节的**程序设计语言**。这里所说的"处理过程细节",不是指待解问题及其解法的本质所要求的(问题逻辑所固有的),而是指为计算机上实现求解任务而设定的计算细节及其执行顺序。

在传统的**高级语言**中,**程序**一般由描述处理对象的一组说明和描述处理动作及执行次序的一组**语句**组成。如果一种语言的基本成分的抽象级较高,那么它的描述能力就较强,在抽象级相对较低的语言中需要用许多语句才能表示的计算过程细节,在抽象级高的语言中往往可以用一条语句加以概括,以更简明直接的形式表示出来。例如,APL 语言中用一条语句可以直接表示两个矩阵相乘的计算任务(如 A←B +. ×C),但在 **FORTRAN 语言**中对同一计算要求,要用三重循环写成的一段程序来表示具体的计算过程。这就是说,语言的抽象级越高,则用它编写的程序中有关计算过程细节的描述就越少,相应地,语言的"非过程性"体现得也更为明显。因此,在一定意义上说,非过程语言的含义是相对的。在许多情况下与其说一种语言是非过程的,不如说语言的过程性特征较少。

目前一般认为非过程语言的主要特征有如下三点:①数据的联想引用机制;②数据的高级操作符;③不指明可变性顺序。联想引用机制是指,不指明访问路径、索引或其他结构位置信息,而只需给出该数据应满足的条件或性质的形式描述就可以存取数据的机制;高级操作符是指可直接表示集合、数组、表等复杂数据运算的操作符;可变性顺序是指改变执行次序也不影响处理结果的顺序,它不是问题求解方法的本质所确定的,而是编写程序时因语言描述能力的限制而程序设计人员加以精化和规定的顺序。

早在 20 世纪 60 年代,就已开始以冯·诺依曼式计算模型为基础的非过程语言的研究。针对数据处理领域中的特定问题,曾出现了一些具有非过程特征的语言,例如:描述判定表的语言 DETAB-65,报表程序的生成器 RPG 等。其中 DETAB-65 不显式指明判定表中规则的执行顺序(可变性顺序),RPG 只需输入关于输出表格信息即可。但这些语言中有关数据运算描述的抽象级较低。而对通用型非过程语言的研究工作,由于它不仅与语言处理技术直接相关,而且与自动程序设计技术也有密切联系,所以进展比较缓慢。迄今为止,设计这类语言时采用的主要办法是,在语言中尽量引进各种抽象度较高的非过程性描述手段,以期做到在程序中增加"做什么"的描述成分,减少"如何做"的细节描述,SETL 和第四代语言(4GL)有其一定代表性。SETL 是以集合论为基础的超高级语言。为了提高程序的抽象级别,它引进集合作为数据类型,并提供"联想

引用”机制和可以直接表示集合运算的很多高级运算符。第四代语言是 20 世纪 80 年代出现的新概念。虽然目前还没有公认的严格定义,但一般认为它是抽象程度很高、非过程性很强的一类语言的统称,例如,各种数据库查询语言、程序生成器和能转换成可执行代码的形式规约语言等,它们一般都具有联想引用机制和高级操作符。

在 70 年代以后出现的**逻辑程序设计语言**和**函数式程序设计语言**是以非冯·诺依曼式计算模型为基础的新型语言。逻辑式程序设计语言的典型代表是 **PROLOG 语言**。语言的基本成分是 HORN 子句,程序则由用它描述的问题求解所需的一系列事实与推理规则组成。程序的执行过程是从给定目标出发,利用程序中的事实和规则,反复进行合一、归结和回溯的过程。然而,这个执行过程基本上没有显式地反映在程序之中。这就是说,PROLOG 语言不仅具有联想引用机制,而且还有不指明可变顺序特征。函数式程序设计语言以函数作为程序的基本成分,并提供一系列用来构造更为复杂函数的定义设施,程序就是根据给定问题使用这些设施构造的求解函数的表达式。由于语言中没有赋值语句,没有副作用,因此如果函数有定义,则计算结果与表达式的计算次序无关。 (金淳兆)

fei jidashi yinshuaji

非击打式印刷机(nonimpact printer) 一种利用物理的(光、电、热、磁)或化学的方法,印字头不与纸或其他媒体接触,或虽有接触但无击打动作的印刷设备。非击打式印刷机有许多类型,它们是基于热敏、喷墨、电灼、静电、电子照相转印以及其他如离子沉积、磁化成像转印等原理制成的。由于这类设备不需击打动作,因而它的工作噪声低,印字速率快。又由于计算机所用的各种非击打式印刷设备都属点阵型,因而字体变换方便,字符种类不受限制,适于汉字印刷,可印图形及图像。

喷墨印刷机 多采用随机喷墨方式,在需要时才从喷嘴喷射墨滴。印字头由一列喷嘴构成,每个喷嘴均可根据需要分别喷出墨滴。产生墨滴的原理有三种:①用压电传感元件的变形压迫压力腔的内壁,从喷嘴挤压出墨滴。②将墨汁加热产生气泡,使墨汁经喷嘴射出。③使用固体色棒,色棒在高压脉冲的作用下溶出墨滴,经喷嘴射出。

利用喷墨技术可实现彩色印刷。在彩色印刷的喷墨印字头上装有青、品红、黄三色及黑色共四套喷嘴和墨盒(照片打印机可多达 7 色),喷嘴个数多的达到 3072 个(512 × 6)。打印分辨率从原来的 180 dpi(点每英寸)提高到后来的 4800 dpi,墨滴体积由以往的 30 pL(1 pL = 10^{-12} L)减小到后来的 2 pL、4 pL。因此,能印出质量非常高的彩色图文。打印速度达到 17 ppm(页每分)(单色,A4)、12 ppm(彩色,A4)。

彩色喷墨印刷机是 2003 年流行的照片印刷工具。在提高印刷质量的同时,喷墨照片印刷机印刷速度达到 1 分钟之内印一张 4 in × 6 in 的照片。

大幅面(A1 以上)彩色喷墨印刷机俗称喷绘机,它在计算机辅助设计(CAD)/地理信息系统(GIS)、广告制作方面有广泛应用。多采用 6 色墨水,2003 年印刷分辨率可达 1 440 dpi,印刷速度可达 720 m^2/h。

电子照相印刷机 利用电子照相转印的印刷设备。在光导电绝缘材料制成的转印鼓(或带)的表面预先施加静电荷,然后,由字符或图形信息调制的光束在其上扫描、曝光。被光照射部分的电荷消失,成像部分的电荷保留下来,形成静电潜像。经过吸附色粉的显影处理后,带电部分吸附上色粉而形成色粉图像。然后,再转印到纸上,热压定影。利用电子照相转印技术制成的印刷设备,因曝光用的光源不同,有激光印刷机、发光二极管印刷机、静电印刷机、液晶快门印刷机、等离子发光管印刷机以及荧光管印刷机等多种类型。其中,激光印刷机的发展历史最久,应用最广。

(1) 激光印刷机 利用激光器件作光源的一种电子照相转印型印刷设备。20 世纪 70 年代中期,西门子公司和 IBM 公司先后都推出了这种印刷机,每分钟可印字符万行以上,属高速印刷设备。其工作原理如下:由激光器发出的激光束经声光调制偏转器按字符点阵的信息调制。在高频超声信号的作用下,声光偏转器衍射出形成字符的调制光束。当频率变化时,激光束的衍射角度随之变化,形成纵向的扇出。此扇出光束经高速恒速旋转的多面镜反射,在预先荷电的、用光电特性材料制成的转印鼓面上沿鼓的轴向扫描曝光。鼓面被激光束照射的部位的电荷消失,形成静电潜像。当鼓面经过带相反电荷的色粉时,由于静电作用吸附上色粉,进行显影。在电晕电场的作用下,色粉由鼓面被转印到纸上。经热滚挤压定影之后,字符便永久性地印在纸上。

激光印刷速度高,印字质量好,分辨率高,噪声小,可印刷多种类型的字符字体、图形、图像,字形大

小变化灵活，适于大量数据的连续印刷。通常激光印刷机以页为单位进行操作，故常称之为页式印刷机。2003 年达到的分辨率为 2 400 dpi，打印速度达 37 ppm（页每分）（黑白，A3）、30 ppm（彩色，A3）。

（2）发光二极管印刷机　利用发光二极管作光源的电子照相转印型印刷设备。发光二极管阵列构成一横排发光点直接在转印鼓或带的表面上曝光，省却了像激光印刷机所用的复杂的光束偏转扫描系统。发光二极管印字头由多片（一般为 30～40 片）发光二极管阵列芯片（每片 64 或 128 个发光点）、驱动集成电路及对应每个发光点的棒状聚焦透镜组成。

由于这种印刷机使用了高集成电路技术，又无可动部件，因而可靠性高，且结构较激光印刷机简单、紧凑。一行 1000～4000 发光点同时沿纸的走向扫描，印字头不需作横向往复运动。分辨率可达 400 dpi，印刷速度为 15～20 ppm（页每分）。

（3）静电印刷机　利用多个电极直接在具有良好绝缘性能的电介质层的特殊纸上施加电荷，形成静电潜像，经过处理而成像的印刷方法。写入电极可以是一横排针或一组扫描针，也可以是针屏阴极射线扫描管上的针屏。当 500～1 500 V 的脉冲电压加到针上时，或当阴极射线管内电子束扫描针屏时，针端便在纸的电介质层上产生电荷，形成静电潜像。然后经过干式或湿式显影及热定影等处理，形成永久性的字符。静电印刷机是一种较早出现的非击打式印刷设备。它的结构简单，噪声小，印刷速度快，每分钟可印 2 万行。但它需用特殊的记录纸，显影与定影过程复杂。

热敏印刷机　利用印字头上多个电热元件在特殊的热敏纸上瞬时加热，引起与热元体接触部位纸的变色而形成字符。热印头上一列或多列热元件的发热点与纸保持适当接触，印头自左至右移动，各热元件在字符点矩阵的控制下快速地热冷交替变换。这种印字机的主要优点是机构简单，体积小，噪声小；其主要缺点是热印头的热反应速率不高，热敏纸的字迹保存性差。

热转印印刷机　利用转印色带将字符转印到纸上的印刷设备。在几微米厚的聚酯薄膜上涂以低熔点的固态墨作为转印色带，一排点状电阻发热元件装在陶瓷基片上构成热印头。印刷时，热印头压在色带与纸上并根据字符点阵适时通过脉冲电流，使色料熔化而转印到纸上。转印色带有许多种，常见的为蜡色带、升华带，其他还有热阻带、染料扩散带等。

热升华印刷机　通过半导体加热元件可调节每一点的温度，组合出色彩的比例和浓淡程度，达到连续色调的彩色照片的效果。转印的画面质量可与银盐相片媲美。但打印速度慢，耗材贵，使用环境要求高。2003 年有的产品达到 334 dpi 的彩色分辨率，印刷一张 3.5×5 cm 的照片的时间是 26 s。

参考文献

Myers R A, Tamulis J C. Introduction to topical issue on non impact printing technologies. IBM J Res Develop, 1984, 28(3): 234-240
（刘锡刚）

feijiandu xuexi

非监督学习（unsupervised learning）　在没有类别信息情况下，通过对所研究对象的大量样本的数据分析实现对样本分类的一种数据处理方法。

由于在很多实际应用中，缺少所研究对象类别形成过程的知识，或者为了判断各个样本（模式）所属的类别需要很大的工作量（例如卫星遥感照片上各像元所对应的地面情况），因此往往只能用无类别标签的样本集进行学习。通过无监督学习，把样本集划分为若干个子集（类别），从而直接解决了样本的分类问题，或者把它作为训练样本集，再用监督学习方法进行分类器设计。常见的非监督学习方法有聚类分析、非监督特征降维技术等。

聚类分析是指将给定数据集划分成几个不相交的非空子集，使得类内的样本相似度最大，类间样本的相似度最小。聚类算法有多种不同的分类方式，如根据是否使用模糊集合，可以分为硬聚类和模糊聚类；根据是否有在线执行的需求，可以分为离线聚类和在线聚类；根据聚类结果表示的不同，可以将聚类算法大致分为四类，基于类原型的聚类算法、层次聚类算法、连通型聚类算法和划分矩阵型聚类算法等。

基于类原型的聚类算法要求每个类可以由其类原型来代表。基本思想是用类原型来代替类中的每个样本时使得误差最小。类原型聚类算法又可根据类原型的不同，分为点类原型聚类算法、超平面原型聚类算法、超球原型聚类算法等。基于类原型聚类算法实现简单，收敛速度快，但是对“噪声”和孤立点数据比较敏感。典型算法有 K-means 算法等。

层次聚类算法是一类用树状结构来表示聚类结果的算法。它可分为凝聚层次聚类算法和分裂层次聚类算法。凝聚层次聚类采用的是自底向上的策

略，首先将样本集的每个样本作为一个类，然后根据相似度的大小对某些类进行合并，形成新的类，不断重复这一过程，直到所有的样本都属于一个类；分裂的层次聚类与凝聚的层次聚类相反，采用一种自顶向下的策略，它首先将整个样本集看作一个类，然后逐渐分裂较大的类为较小的类，重复这一过程直到最后每个样本成为一个类。典型的凝聚层次聚类算法有 Single Linkage，Average Linkage，Complete Linkage 等。

连通型聚类算法是一种使用图的连通分支来表示类的聚类算法。典型的连通型聚类算法有最小生成树（MST）算法、最小切聚类算法（minimum cut）、DBSCAN、DENCLUE 等。

划分矩阵型聚类算法用划分矩阵（包括软划分）来表示聚类的结果。常见的算法有基于矩阵分解技术的算法、基于信息论的方法以及基于 margin 理论直接对数据进行无监督标定的方法。

非监督特征降维是一种将给定数据集从已知的高维特征空间表示导出其低维特征空间表示的方法，可分为线性降维和非线性降维。线性降维包括主成分分析（PCA）、奇异值分解等方法。非线性降维主要有流形学习、非线性成分分析（NLCA）、独立成分分析（ICA）、多维尺度变换（MDS）、自组织特征映射（self-organizing feature map），以及非负矩阵分解等。

参考文献

1. Duda R O, Hart P E, Stork D G. Pattern classification. 2nd ed. John-Wiley & Sons, 2001

2. Ghahramani Z. Unsupervised Learning. Advanced Lectures on Machine Learning LNAI 3176. 2004

（于剑）

feixianxing daishu fangchengzu shuzhi jiefa

非线性代数方程组数值解法（numerical solution for system of nonlinear algebraic equations） 求非线性代数方程组

$$f_i(x_1,x_2,\cdots,x_n)=0,\quad i=1,2,\cdots,n \tag{1}$$

数值解的方法。式(1)中 $f_i(x_1,x_2,\cdots,x_n)$ 是定义在 n 维空间 R^n 的开域 D 上的实函数，且至少有一个是关于 $x_1,x_2,\cdots,x_n$ 的非线性函数。在 R^n 中记 $\boldsymbol{x}=(x_1,x_2,\cdots,x_n)^{\mathrm{T}}$，$\boldsymbol{f}=(f_1,f_2,\cdots,f_n)^{\mathrm{T}}$，则方程组(1)简写为 $\boldsymbol{f}(\boldsymbol{x})=0$. 若存在 $\boldsymbol{x}^*\in D$，使得 $\boldsymbol{f}(\boldsymbol{x}^*)=0$，则称 $\boldsymbol{x}^*$ 是非线性方程组的解。非线性方程组可能有一个解或多个解，也可能有无穷多解或无解。除极特殊方程外，一般不能用直接法求得其精确解。通常只能采用迭代法求满足精度要求的近似解。迭代法就是根据不同思想构造逐次逼近方程组的解 $\boldsymbol{x}^*$ 的迭代序列

$$\boldsymbol{x}^{k+1}=\boldsymbol{g}(\boldsymbol{x}^k),\quad k=0,1,\cdots \tag{2}$$

或

$$\boldsymbol{x}^{k+1}=\boldsymbol{g}(\boldsymbol{x}^k,\cdots,\boldsymbol{x}^{k-m+1}),\quad k=0,1,\cdots \tag{3}$$

前者称为单步迭代法，后者称为多步（m 步）迭代法。计算时只要给定初始近似 $\boldsymbol{x}^0$，则可由式(2)逐次计算出 $\boldsymbol{x}^1,\boldsymbol{x}^2,\cdots$。当 $\lim_{k\to\infty}\boldsymbol{x}^k=\boldsymbol{x}^*$ 时，则称迭代序列 $\{\boldsymbol{x}^k\}_{k=1}^{\infty}$ 收敛于 $\boldsymbol{x}^*$。如果存在常数 $p\geqslant 1$ 以及 $c_p>0$，使得

$$\lim_{k\to\infty}\frac{\|\boldsymbol{x}^{k+1}-\boldsymbol{x}^*\|}{\|\boldsymbol{x}^k-\boldsymbol{x}^*\|^p}=c_p,\quad c_1<1$$

则称迭代序列 $\{\boldsymbol{x}^k\}_{k=1}^{\infty}$ 对于 $\boldsymbol{x}^*$ 是 p 阶收敛的，c_p 称为收敛因子。$p=1$ 为线性收敛，$p>1$ 称为超线性收敛。收敛阶数 p 越大，收敛越快，在 p 相同的情况下，则 c_p 越小收敛越快。

求解非线性方程组最重要的迭代法有牛顿法及其变形、拟牛顿法等。

牛顿法及其变形 牛顿法基本思想是将非线性函数 $\boldsymbol{f}(\boldsymbol{x})$ 逐步线性化形成以下的迭代序列：

$$\boldsymbol{x}^{k+1}=\boldsymbol{x}^k-\left[\frac{\partial\boldsymbol{f}(\boldsymbol{x}^k)}{\partial\boldsymbol{x}}\right]^{-1}\boldsymbol{f}(\boldsymbol{x}^k),\quad k=0,1,\cdots \tag{4}$$

其中

$$\frac{\partial\boldsymbol{f}(\boldsymbol{x}^k)}{\partial\boldsymbol{x}}=\begin{bmatrix}\frac{\partial f_1(\boldsymbol{x}^k)}{\partial x_1} & \frac{\partial f_1(\boldsymbol{x}^k)}{\partial x_2} & \cdots & \frac{\partial f_1(\boldsymbol{x}^k)}{\partial x_n}\\ \frac{\partial f_2(\boldsymbol{x}^k)}{\partial x_1} & \frac{\partial f_2(\boldsymbol{x}^k)}{\partial x_2} & \cdots & \frac{\partial f_2(\boldsymbol{x}^k)}{\partial x_n}\\ \vdots & \vdots & & \vdots\\ \frac{\partial f_n(\boldsymbol{x}^k)}{\partial x_1} & \frac{\partial f_n(\boldsymbol{x}^k)}{\partial x_2} & \cdots & \frac{\partial f_n(\boldsymbol{x}^k)}{\partial x_n}\end{bmatrix}$$

是 $\boldsymbol{f}(\boldsymbol{x})$ 的雅克比矩阵。当 $\boldsymbol{x}^0$ 是解 $\boldsymbol{x}^*$ 的一个较好近似时，牛顿迭代序列(4)是 2 阶收敛的。由 $\boldsymbol{x}^k$ 计算 $\boldsymbol{x}^{k+1}$ 的步骤为：

(1) 计算 $\boldsymbol{f}(\boldsymbol{x}^k)$ 及 $\frac{\partial\boldsymbol{f}(\boldsymbol{x}^k)}{\partial\boldsymbol{x}}$；

(2) 采用直接法求解线性方程组 $\frac{\partial\boldsymbol{f}(\boldsymbol{x}^k)}{\partial\boldsymbol{x}}\Delta\boldsymbol{x}^k=-\boldsymbol{f}(\boldsymbol{x}^k)$，此方程组称为牛顿方程；

(3) 计算 $\boldsymbol{x}^{k+1}=\boldsymbol{x}^k+\Delta\boldsymbol{x}^k$。

迭代停止准则为

$$\| x^{k+1} - x^k \| \leqslant \varepsilon \quad 或 \quad \| f(x^k) \| \leqslant \varepsilon$$

其中 ε 为给定精度。牛顿法的优点是收敛速度快且可以自修正,缺点是每步要计算雅克比矩阵。另外,要求x^0 在x^*附近较难达到。

为放宽牛顿法对初始近似x^0 的限制,扩大收敛范围,可引进松弛参数 $\omega_k>0$,将牛顿法程序改为

$$x^{k+1} = x^k - \omega_k \left[\frac{\partial f(x^k)}{\partial x}\right]^{-1} f(x^k), \quad k = 0,1,\cdots \tag{5}$$

称为牛顿下山法。

为减少牛顿法计算工作量,可用修正牛顿法。为了避免直接计算雅克比矩阵,可以采用插商近似偏导数的离散牛顿法。修正牛顿法和离散牛顿法均具有较快的收敛速度。

在牛顿法(4)中,若采用解线性方程组的迭代法求解牛顿方程组,则得到不精确牛顿法。不精确牛顿法是一个内外迭代过程。如果采用求解线性方程组的 Krylov 子空间方法求解牛顿方程,则所得方法就是牛顿 Krylov 子空间方法。在牛顿 Krylov 子空间方法中,与雅克比矩阵相关的运算均为矩阵向量乘积 $\frac{\partial f(x^k)}{\partial x}v$,其中 v 为 Krylov 子空间向量。雅克比矩阵向量乘积可用有限差分

$$\frac{\partial f(x^k)}{\partial x}v \approx \frac{f(x^k + \sigma v) - f(x^k)}{\sigma}$$

加以近似,其中 σ 为差分步长。于是,在牛顿 Krylov 子空间方法中,可以不用形成和存储雅克比矩阵。称此类方法为无雅克比矩阵的牛顿 Krylov (Jacobian-free Newton-Krylov, JFNK) 子空间方法。该类方法适合求解大规模问题。

拟牛顿法 一类不用计算雅克比矩阵,又具有超线性收敛速度的算法。该类算法出现于 20 世纪 60 年代中期,有很多不同的计算公式,其中常用的有秩 1 拟牛顿法以及秩 2 拟牛顿法。

此外,求解非线性方程组的方法还有割线法、布朗方法、布兰特方法、连续法、区间方法和多分裂算法等多种方法,其中布兰特方法由于效率较高,已在数学包中被广泛使用。连续法由于对初始近似x^0没有特别限制而受到重视。区间方法在判断非线性方程解的存在唯一性方面较简便。多分裂算法适用于并行计算,近年有很大发展。

在方程(1)中当 $n=1$ 时,就是非线性方程 $f(x)=0$,前面介绍的牛顿法、牛顿下山法、离散牛顿法、割线法等都是常用的数值方法,其中割线法又称弦位法,其迭代公式为

$$x_{k+1} = x_k - \frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})} f(x_k), \quad k = 1,2,\cdots \tag{6}$$

这是一种多步迭代法,计算时先给出初始近似 x_0, x_1,再按公式(6)逐次求出 x_2, x_3,…。割线法每步只算一个新函数值,计算量较省,而方法收敛阶为 $p=(1+\sqrt{5})/2=1.618\cdots$,故效率也较高。

参考文献

1. 李庆扬,莫孜中,祁力群. 非线性方程组数值解法. 北京:科学出版社,1992

2. Kelley C T. Iterative methods for linear and nonlinear equations. SIAM, Philadelphia, 1995

3. Kelley C T. Solving nonlinear equations with Newton's method. SIAM, Philadelphia, 2003

(李庆扬 安恒斌)

feiyishi xinxing bandaoti cunchuqi

非易失新型半导体存储器(new types of non-volatile random access memory) 通电时、断电后均能保持存储数据的固态存储器。新型的半导体的非易失存储器包括闪存(参见**快可擦编程只读存储器芯片**)、**相变随机存储器**、阻变随机存储器、铁电随机存储器、磁变阻随机存储器等。

相变随机存储器(phase change random access memory) 一种利用相变材料在有序结构(晶态)与无序结构(非晶态)下所表现出的电阻率极大差异以及两种状态间的可逆转化特性来存储信息的随机存储器。相变材料在晶态具有较低电阻,非晶态具有较高电阻。相变存储器存储单元结构如图 1 所示。相

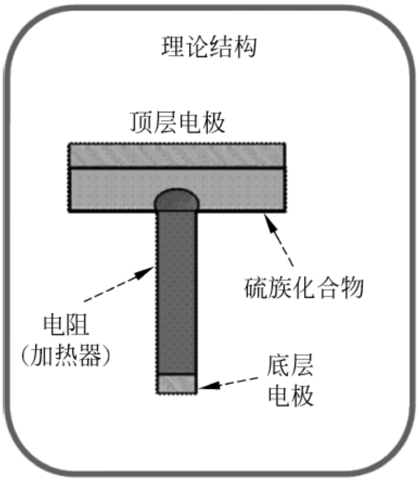


图 1 相变存储器存储单元结构

变存储单元的状态改变是通过在顶层电极与底层电极间加入电流产生局部焦耳热来加热相变材料实现的。当要往存储单元写入状态“0”(非晶态)时,在顶层电极与底层电极间注入一个强而窄的电流脉冲,先将相变材料加热到熔融状态然后快速冷却,使相变材料达到非晶态。当需要写入状态“1”(晶态)时,在电极间注入一个弱而宽的电流脉冲,将相变材料加热到熔点以下、晶化温度以上,使得相变材料逐步结晶,到达晶态。写入“0”或“1”通过控制电流脉冲的强度与时间长度来实现。读取存储单元的数据则通过测量存储单元的电阻值来实现。

根据每个存储单元保持数据量的不同,相变存储器分为单阶单元(SLC)与多阶单元(MLC)。在SLC中,每个存储单元只存在高阻值与低阻值的差别;在MLC中,通过控制电流脉冲的长度,实现部分结晶态,达到2^n不同结晶状态,即2^n不同阻值,因此可以保存n位数据。

阻变随机存储器(resistive random access memory) 用忆阻器(memristor)实现的随机存储器。忆阻器是一种具有记忆功能的非线性电阻,它使用强相关电子类的材料(例如过渡金属氧化物、钙钛矿结构氧化物等),可通过电流的变化控制阻值的变化:从一个方向对材料施压电压脉冲可使材料成为高阻值,从另一个方向施加电压脉冲则会使材料转变成低阻值。运用阻值高、低的两种状态可以储存数据。

在阻变随机存储器的每个存储单元中,有一根导线与存储材料的一边接触,另一根导线与存储材料的另一边接触。存储单元是一个由两个金属电极夹着的存储材料层构成的双端、双层交叉开关结构的半导体。通过检测交叉开关两端电极的阻性,就能判断存储单元的“开”或者“关”状态。

根据不同材料和结构,阻变随机存储器的存储机理有多种解释,包括导电丝模型、界面模型、陷阱电荷模型、边界迁移模型等,其物理机制的不清楚一直制约着阻变随机存储器的实用化。

铁电随机存储器(ferroelectric random access memory) 一种利用铁电晶体的铁电效应实现数据存储的随机存储器。铁电效应是指在铁电晶体上施加一定的电场时,晶体中心原子在电场的作用下运动,并达到一种稳定状态;当电场从晶体移走后,中心原子会保持在原来的位置。如图2所示,当一个电场被施加到铁晶体管时,中心原子顺着电场停在低能量状态的位置(下左),反之,当电场反转被施加到同一铁晶体管时,中心原子顺着电场的方向在晶体里移动并

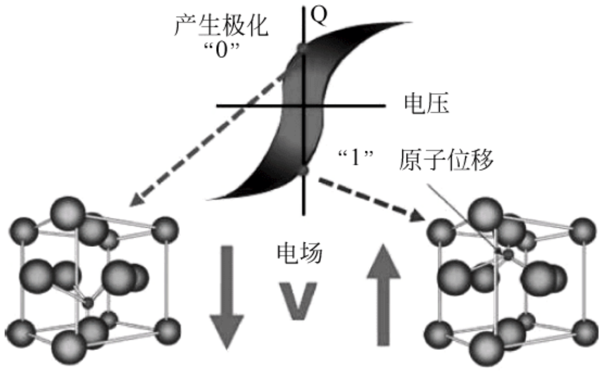


图2　铁电效应示意图

停在另一低能量状态(下右)。大量中心原子在晶体单胞中移动耦合形成铁电畴,铁电畴在电场作用下形成极化电荷。铁电畴在电场下反转所形成的极化电荷较高,铁电畴在电场下无反转所形成的极化电荷较低,这种铁电材料的二元稳定状态使得铁电可以作为存储器。

当移去电场后,中心原子处于低能量状态保持不动,存储的状态也得以保存不会消失。铁电畴的反转不需要高电场,仅用一般的工作电压就可以改变存储单元是在“1”或“0”的状态,也不需要电荷泵来产生高电压擦除数据,因而没有擦写延迟的现象。

磁变阻随机存储器(magnetoresistive random-access memory) 一种利用磁电阻效应和电子自旋现象进行数据存储的随机存储器。磁变阻随机存储器的每个存储单元包含三层结构,其中有两个磁性层,而在磁性层之间存在一层厚度为纳米级的非磁性中间层。当上下两个磁性层的磁化现象互为平行时,电阻会较相反时小。通过对这个三层结构的控制电路施以不同方向的电流,并且电流大小超过磁性层的矫顽力,改变磁性层的方向,就可以记录“0”和“1”。而读出过程不会改变磁层电子的自旋方向,因此读操作是非破坏性的。

自旋转移矩随机存储器(spin torque transfer random access memory, STT-RAM) STT-RAM是基于MRAM技术的新型存储器,其工作原理不同于MRAM中利用电流产生的磁场进行信息写入的模式,而是利用电流本身进行信息的写入。STT-RAM的结构和垂直各向异性MRAM的结构基本相同——在Si或SiO_2衬底上沉积出垂直隧道结三明治结构,由上下两个磁性层和一个非磁性非金属隧道隔离层组成,如图3所示。

两个磁性层均由高垂直各向异性层和自旋极化加强层组成。具有高垂直各向异性的磁性层通过交

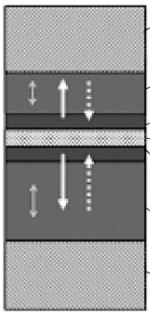


图 3　STT-RAM 的存储单元

换耦合作用,将相邻的自旋极化加强层的磁化方向沿垂直方向固定。工作时,电流流过存储单元,电子会在穿过自旋极化加强层的时候被极化,从而产生高度极化的电流。极化电流产生的力矩作用于自旋极化加强层的磁矩,使其翻转;而实际上由于两个自旋极化加强层的矫顽力不同,只有上层矫顽力较小的膜层的磁矩会发生翻转,成垂直向上或向下排列。而通过控制电流方向,可以控制磁矩的翻转方向,从而控制隧道结两端的磁性层的磁矩的排列模式——平行或反向平行排列,不同排列模式的隧道结结构会表现出不同大小的磁阻,从而达到写入 0 和 1 的目的。

从 STT-RAM 的工作原理可以看出,在其存储单元的信息写入过程中,两磁性层磁矩的相对方向由通过存储单元的电流方向决定,而不是由电流产生的磁场(如 MRAM)决定,其极化电流所产生的力矩的大小只和电流密度成正比。

参考文献

1. Lee B C, Ipek E, Mutlu O, et al. Architecting phase change memory as a scalable DRAM alternative. Proceedings of ISCA09, 2011

2. 徐迎晖. 磁电阻随机存储器 MRAM 的原理与应用. 电子技术,2006,3

3. 杨玉超,等. 新一代超快高密 Ag/ZnO:Mn/Pt 非易失性阻变存储器及其工作机理. 中国真空网,2010

4. Zhang Shuchao, et al. Simulation study of new 3-terminal devices for high speed STT-RAM. Journal of Semiconductors, 2011, 32(7)　　(缪向水　冯丹)

fei zhenshigan tuxing huizhi

非真实感图形绘制(non-photorealistic rendering,NPR)　相对**真实感图形生成**而提出来的绘制方法。其他如 expressive rendering, artistic rendering, painterly rendering, interpretative rendering 等是 NPR 的同义词,最近 stylised rendering 即风格化绘制逐渐被人们所接受。

早期的非真实感图形绘制(NPR)技术主要是模拟传统绘画效果,如钢笔画、铅笔画、油画、水彩画等,近年来出现一些 NPR 技术绘制出的图形既有别于真实感也有别于传统绘画效果。在 NPR 技术分类上有基于模型的、基于图像的以及两者混合的,也可以从 NPR 技术应用到二维或是三维空间以及是否需要交互进行分类。

NPR 技术主要有两部分内容:

(1) 笔刷生成　它包括笔刷纹理和笔刷形状,对于传统笔刷纹理可以用模型模拟,也可以用图像合成。在模拟传统笔刷纹理时往往需要考虑构成画笔材料的特性、绘画颜料的特性以及画纸或画布的特性。笔刷形状可以用施加在画笔上力的大小作为输入参数进行计算,也可以用固定函数定义。在一些非传统笔刷模型中,其纹理和形状可以有很多变化,如圆弧、螺旋线以及字母等都可以作为笔刷纹理单元。

(2) 笔刷绘制　包括控制笔刷的方向、长短、粗细以及在二维平面或三维曲面上的分布。在不同技术种类及不同应用中对它们有不同的控制方法。

在基于模型的二维 NPR 技术中需要在二维平面上设计画笔的运动轨迹,轨迹的方向即为笔刷的方向,笔刷的长短、粗细则根据应用需要设定。在基于图像的交互 NPR 技术中,笔刷方向通过用户交互指定,一般交互量越大,所绘制的效果越接近手工绘画。在基于图像的自动 NPR 技术中,一般需要提取物体的边缘来提供笔刷方向信息,物体区域中的笔刷方向要通过对边缘信息进行某种运算来确定。为了强化手工绘画效果,可以采用多层处理技术进一步变化笔刷的长短和粗细。

在基于模型的三维自动 NPR 技术中根据笔刷长短有不同的处理方法。对于长笔刷一般先检测物体边缘及物体表面褶线,然后再进行笔刷绘制。对于短笔刷(包括非传统笔刷)则在物体网络内分布,其位置进行局部随机扰动。在物体宏观处理上,笔刷的分布密度可结合光照模型计算出来的明暗进行控制。在基于模型的三维交互 NPR 技术中,用户可以选择多种预先设计的笔刷纹理绘制到物体边缘和表面上。

NPR 技术种类繁多,绘制出的效果风格各异。

基于这种特点，NPR 技术仍在继续发展中。

参考文献

1. Non-photorealistic rendering. Siggraph'99 Course 17, August, 1999

2. Robert D, et al. WYSIWYG NPR: Drawing Strokes Directly on 3D Models. ACM Transactions on Graphics (TOG), 2002, 21(3): 755-762 （于金辉）

fenbie bianyi

分别编译（separate compilation） 在一个较大的程序编译过程中，对组成该程序的多个高级语言代码模块文件分别独立进行编译，生成多个浮动目标程序，供以后链接程序组合成最终的目标程序之用。

早期程序开发中，源程序往往放在一个文件中，编译程序将源程序文件整体地直接翻译成目标机器可执行的目标代码。随着软件中分而治之思想与模块化设计原则的出现，程序设计语言支持将程序组织成多个文件，原来的翻译过程被细划分为编译、汇编和链接三个阶段。这形成的一个重要变化是各部分程序文件不必整个一起同时编译，而是分别编译。编译程序与汇编程序一起将程序模块生成可重定位的浮动目标代码，然后通过链接程序将多个目标代码组合成最终目标程序。

现代程序设计语言和编译器都支持程序的分别编译，在实现技术上支持了软件工程中模块化设计原则和代码可重用的最大化原则，对于提高程序的开发效率、优化编译效果均有重要作用。 （黄春）

fenbu canshu xitong fangzhen

分布参数系统仿真（distributed parameter system simulation） 面向用偏微分方程描述的一类系统的仿真技术（参见**计算机仿真**）。

随着现代科学技术的发展，人们正在不断建造更为快速的交通工具、更大规模的建筑物、更大跨度的桥梁、更大功率的发电机组和更为精密的机械设备。这一切都要求工程师在设计阶段就能精确地预测出产品和工程的技术性能，需要对结构的静、动力强度以及温度场、流场和电磁场等技术参数进行分析计算。例如分析计算高层建筑和大跨度桥梁在地震时所受到的影响，判断是否会发生破坏性事故；分析计算核反应堆的温度场，确定传热和冷却系统是否合理；分析涡轮机叶片间的流体动力学参数，以提高其运转效率。这些分布参数系统仿真都可归结为求解偏微分方程，主要有三类解法：差分法、线上求解法以及有限元法。

差分法是偏微分方程仿真的经典方法，其原理是：在空间和时间两个方面将系统离散化，从而将偏微分方程化为差分方程，然后从初值或边值出发，逐层推进。它具有高度的通用性，易于程序实现。用一个差分方程近似一个偏微分方程时，必须注意差分格式的相容性、收敛性及稳定性。一般说来，用一个差分模型来近似微分模型，相容性条件必须满足。如果误差始终保持在一定的范围，而不是越来越大，则称该差分方法具有稳定性。对于一个适当的偏微分方程初值问题，如果逼近它的差分格式是和它相容的，且具有稳定性，则该差分模型收敛于原偏微分方程，我们称该差分方法具有收敛性。

另一类方法是**线上求解法**（method on lines）。这是 20 世纪 70 年代以来，基于常微分方程仿真方法发展起来的另一大类偏微分方程仿真建模方法。线上求解法的基本思想是：将空间变量进行离散化，而时间变量仍保持连续，从而将偏微分方程转化为一组常微分方程，进而基于常微分方程的各种仿真算法对时间离散化进行仿真。仿真过程中，基于时间数值积分与空间差分交替进行。在使用这种方法时，正确选择差分方法以实现对空间变量求导，这是保证仿真模型稳定性及计算精度的前提。

有限元法（finite element method, FEM）的基本思想是将连续的求解区域离散为一组有限个、且按一定方式相互连接在一起的单元的组合体。由于单元能按不同的连接方式进行组合，且单元本身又可以有不同的形状，因此可对几何形状复杂的问题建立模型并进行求解。有限元法作为数值积分法，一个重要的特点是利用在每一个单元内假设的近似函数来分片地表示全求解域上待求的未知场函数，而单元内的近似函数通常由未知场函数或其导函数在单元的各个节点的数值和其插值函数来表达，从而使一个连续的无限自由度问题变成离散的有限自由度问题，有限元法因此而得名。一经求出未知场函数或及其导函数在单元的各个节点上的数值，就可以通过插值函数计算出各个单元内的场函数的近似值，从而得到整个求解域上的近似解。随着单元数增加及插值函数的精度提高，且单元满足收敛要求，近似解收敛于精确解。有限元方法较之采用直交网格的差分法具有独特的优越性，易于适应区域形状的任意性，也易于用计算机程序实现，特别是在高性

能计算机支持下,有限元方法得到广泛的应用。

参考文献

1. 冯康,等. 数值计算方法. 北京:国防工业出版社,1978

2. 熊光楞. 数字仿真算法与软件. 北京:宇航出版社,1991

3. 王瓞成,邵敏. 有限单元法基本原理和数值方法. 2 版. 北京:清华大学出版社,1997

(肖田元)

fenbu jisuan zhongjianjian

分布计算中间件(distributed computing middleware) 在网络环境中,建立在具有基本通信协议的操作系统之上,支持分布式应用软件的有效开发、部署、运行和管理的支撑软件。分布计算中间件是一种起承上(应用软件)启下(操作系统)作用的支撑软件,它支持一体化网络计算,故又称为网络计算中间件,或称软件中间件,简称中间件。

沿　革

在网络应用需求的带动下,中间件于 20 世纪 80 年代中后期形成,并在 90 年代得到蓬勃发展。80 年代,基于 TCP/IP 协议的进程间通信机制及其**套接字**编程接口是用于开发网络应用的主要手段。与 60 年代前用**低级语言**编写程序一样,用套接字编程接口开发分布式应用十分烦琐、复杂和低效。人们开始研究便于分布式应用开发、部署、运行和管理的分布计算中间件。Tuxedo 是早期的中间件,主要用于解决分布式交易事务中的监控问题,它把复杂的监控管理从应用软件中分离出来,从而有效提高了分布式应用软件的开发效率。进入 90 年代,随着互联网及其应用的发展,以支持异构环境中的应用互操作为主要目的的分布计算中间件得到迅速发展,并很快被人们称为继操作系统和数据库管理系统之后的又一类基础软件,已成为网络应用软件系统集成的有力工具。21 世纪以来,面向应用领域的平台化中间件成为人们关注的焦点。

基本内容

中间件种类繁多。从通信机制和程序设计风范的角度看,中间件可以分为:①远程过程调用中间件　支持客户-服务器模式的分布式应用之间采用一种请求-响应的同步方式进行通信,是过程式程序设计风范在分布式应用中的扩展;②消息中间件　支持分布式应用之间采用消息传递的同步/异步方式进行通信和数据交换,通过队列管理、路由管理和发布-订阅等服务,将基于包交换的低层通信机制作了自然扩展;③面向对象中间件　支持面向对象的多层客户-服务器模式的分布式应用,客户通过本地的对象请求代理访问异地服务器上的对象,是面向对象程序设计风范在分布式应用中的扩展,CORBA,JEE 和.NET 是目前面向对象中间件的三个有代表性的技术体系;④面向服务中间件　支持分布式应用通过互联网上的 Web 服务技术实现跨域的信息通信,通常适用于部门与部门或行业与行业间松耦合、粗粒度的资源共享和集成,为开创面向服务的程序设计风范提供了有力支持。

随着互联网技术的不断发展和中间件的广泛应用,中间件的内涵和外延不断拓展。从中间件的功能及其应用的层次看,中间件可以分为基础中间件和应用集成中间件两大类。

(1) 基础中间件的作用主要是对网络上各种软硬件资源进行调度和管理,并提供网络应用所必需的公共服务和各类构件的运行环境。基础中间件的核心是微内核集成总线,通常还有必要的各种基础性服务和运行环境,其各组成部分的主要功能如下:①微内核集成总线　旨在屏蔽底层各种异构的网络和操作系统,在物理位置透明的情况下支持异地构件与服务之间的互访,实现软件模块的即插即用;②通信服务　为各种客户提供访问接口和访问协议,支持客户对服务器端应用业务逻辑的访问,通信协议通常包括 JRMP,SOAP,IIOP,HTTP/HTTPS 等;③公共服务　提供名录、日志、时间、消息、事件、通告、安全、负载均衡、数据访问、事务等共性基础服务;④构件容器　提供 EJB,Web,CCM 等实体构件、会话构件、过程构件和服务构件的运行环境;⑤业务引擎　用于启动相关构件和服务,如 Portal, BPEL, BPMN 等;⑥基本工具　包括各种应用程序的接口定义与编译工具以及各种建模、开发、定制、部署、组装和构件管理工具。搭建在微内核集成总线上的基础中间件又称为应用服务器。

(2) 应用集成中间件内容丰富、自成体系,如企业应用集成中间件,产品化的中间件往往以套件形式呈现,且可拆分为数据、代码、流程、门户、服务等几种集成中间件,分别支持应用软件系统资源管理层、业务逻辑层和应用表现层的集成。

数据集成中间件的任务是支持应用软件数据资源的采集、抽取、转换、包装与传输,把不同来源、格式、性质的各种异构数据源进行物理集成或逻辑集

成,建立满足 QoS 要求的归一化数据访问服务。为此,该中间件向下需协调各数据库系统,向上应为集成数据的应用提供统一数据模式和数据访问的通用接口。如何解决好数据的异构性、完整性和语义冲突的问题,是该中间件技术的关键。基于 XML 的数据集成中间件已在实现异构数据的集成中得到广泛应用。

代码集成和业务流程集成中间件用于支持业务逻辑层应用程序的集成。其中,应用代码集成中间件利用适配机制把新建和遗留应用代码中各类方法统一成标准的应用接口,并包装为消息的形式,通过消息代理机制,支持应用代码的互联互通互操作;业务流程集成中间件对业务流程的整个生命周期进行管理和控制,以组织和协调参与流程的各应用代码之间的动态执行关系,并监控流程的执行状况,该中间件通常提供可视化开发方法,应用开发者通过流程图便可定制或修改业务流程,从而达到流程控制零编程的目的。

门户集成中间件是针对应用软件系统的应用表现层构建的,其作用是支持不同的应用需求,通过调用业务逻辑层中的程序代码和资源管理层中的信息资源,为不同角色的用户提供个性化服务。功能一般包括:集中的门户管理和开发,个性化的内容组织与管理,单点认证登录以及统一而直观的用户界面和各类图表的自动生成等。

服务集成中间件是互联网环境下基于 Web 服务技术支持跨域集成的中间件。通过 Web 服务包装,以及语义 Web、服务组合、服务协同、服务发布、智能搜索和服务发现等技术,该中间件支持 Web 服务成为构建各类应用软件的基本单元,从而支持应用软件开发者无须进行底层的程序实现,而只需编写某些组合脚本便可构建一个复杂的业务应用。该中间件是目前研究较活跃的中间件。

中间件一般属通用软件平台。随着应用的不断深入,目前呈现领域化的发展趋势。在某些重要的应用领域,通过进一步提取具体应用领域中的共性业务需求,将相关解决方案服务化、构件化,建立适用于该领域应用的技术架构,这种融入应用解决方案的领域中间件也称为领域应用框架。该框架为构建网络应用软件提供了综合集成平台以及各种通用和专用的构件库,并支持进一步搭建系统顶层设计所需的整体技术架构。

展　望

在中间件领域化的发展趋势下,当前值得关注的是面向云计算和物联网的中间件。随着以云计算和物联网为核心的信息网络技术以及软件服务工程的快速发展,中间件迎来了重大发展机遇,也给中间件提出了许多新的挑战性课题。这些课题涉及网络资源管理、任务管理、用户管理和安全管理,主要有:如何充分运用多层次多方位的虚拟化技术,支持云计算成为集中式同构无限可扩展的网络计算;如何通过非关系数据库和弹性缓存等技术,有效支持云计算系统高可用、可伸缩的海量数据存储与快速检索;如何通过资源按需聚合机理、资源按需使用的弹性伸缩技术和面向服务的体系架构,支持云计算平台实现软件即服务乃至一切皆服务的按需自助服务;如何在多个层次多个租户之间灵活支持云计算在公用计算环境下安全可信的效用计算;如何面对来自大量不同传感器数据源且不断增长的不确定海量时序关联数据和相关的海量传感器事件,对海量传感器数据和复杂事件进行有效的实时处理;如何运用自适应的软件体系结构和普适计算中的共性支撑服务,支持物联网的多网融合环境和泛在化的应用需求;如何运用数据挖掘和业务智能等人工智能技术,支持物联网对物理世界的深度感知等。所有这些课题亟待人们去探索,去攻克。

参考文献

Geihs K. Middleware challenges ahead. Computer, 2001: 24-31

(吴泉源　王怀民)

fenbu jiaohushi fangzhen

分布交互式仿真(distributed interactive simulation,DIS) 指在高速计算机网络支持下,将分布在不同地点的仿真系统集成起来,通过仿真实体间的实时数据交换,在时空一致的人机交互仿真环境中并行进行仿真的一种先进仿真技术(参见**计算机仿真**)。

分布交互式仿真是**并行仿真**网络化思想的发展。尽管在 20 世纪 80 年代初就出现了多台计算机联网仿真,并作为一种松连接的并行仿真技术加以使用,但是,随着**计算机网络**技术的迅速发展,网络传输速度大大提高,而多处理机系统无论其硬件技术还是软件技术(特别是并行操作系统和并行语言)却步履缓慢,加上仿真技术所面对的问题规模日益庞大,种类也日趋增多,于是人们重新着力于分布式仿真技术的研究。

分布交互式仿真系统的结构特点　分布交互式仿真系统从结构上可以视为由仿真节点和计算机网络

组成。所谓仿真节点本身可以是一台独立的仿真计算机，甚至是一个仿真系统。与单独仿真计算机不同，DIS的仿真节点不仅要完成本节点的仿真任务（如运动学、动力学模型的计算，人机交互，仿真图形或动画的生成等），还要负责将本节点的有关信息发送到其他节点；同时，它也必须按一定周期接收其他节点发送来的信息，并作为执行本节点任务时的输入或条件。一般来说，DIS中的节点往往是任务各异的，这要根据仿真系统的任务分配而定；而且，也并不要求计算机的类型、操作系统类型、编程语言类型一定相同。为了保证这种异构系统的信息共享与通信，需要在应用层定义信息交换的格式和内容。这些交换的信息单元称为协议数据单元（protocol data unit，PDU），使得交换的信息内容与通信硬件和软件无关，从而，易于采用通用的商业网络软硬件环境（如TCP/IP协议）。

DIS除了在功能上具有分布性以外，在地域上往往也是异地分布的。根据其地理分布的广度，计算机网络也可采用多种形式。如果相对比较集中，则可采用**局域网**（local area network，LAN），如果分布比较分散，则需采用广域网（wide area network）或通过网桥（bridge）、路由器（router）或**网关**（gateway）等互连设备构成网络。一种典型的分布交互式仿真系统结构如图1所示。

分布交互式仿真技术特点 分布交互仿真与以往的仿真技术不同，具体表现在以下方面：在体系结构上，由过去的集中式、封闭式发展到分布式、开放式和交互式，构成互操作、可移植、可伸缩及强交互的协同仿真体系结构；在功能上，由原来的单个武器平台的性能仿真，发展到复杂环境下，以多武器平台为基础的体系与体系对抗仿真；在手段上，从单一的构造仿真、实况仿真和虚拟仿真，发展成集上述多种仿真为一体的综合仿真系统；在效果上，由只能从系统外部观察仿真的结果或直接参与实际物理系统的测试，发展到能参与到系统中，与系统进行交互作用，并可得到身临其境的感受。

分布交互仿真是计算机技术的进步与仿真需求不断发展的结果，其特点主要表现在5个方面，即分布性、交互性、异构性、时空一致性和开放性。

（1）分布性 分布性包含功能及地理位置两方面的分布。地理位置上的分布性通过网络实现其集成，而功能分布性表现在：以往的仿真系统使用一个中央计算机来维护整个虚拟世界的状态，并计算每个实体的行为对其他实体和环境的影响；在分布交互仿真系统中没有中央处理机，甚至没有中央计算机控制整个仿真活动，各仿真节点是平等的。每个仿真节点具有各自的资源，并负责对分配给它的仿真任务进行处理。DIS中节点一般具有自治性，即：一个节点的加入与退出不会影响其他节点正常执行。实际上，大多数DIS不是执行单一仿真任务的系统，而往往是将功能各异的若干独立仿真器集成起来以完成更大规模的系统仿真。

（2）交互性 DIS的交互性包括：①节点内部进程交互、人机交互；②节点之间的交互作用；③不仅人机交互及节点之间的交互要求实时性，而且要保证空间的一致性，我们称其为时空一致性。时空一致性是DIS的基本要求，也是DIS的本质特征。

（3）异构性 分布式交互仿真的另一个突出特点是对已有系统的集成。如何将这些地域上分散、不同的制造厂商开发、系统的硬件和软件配置各不相同、实体表示方法与精度各异的仿真节点连接起来并实现互操作，就成为研究人员待解决的问题，也是DIS技术中的关键。

（4）时空一致性 在分布式交互仿真系统中，人是通过计算机生成的综合环境的各种真实的感受来做出响应而形成“人在回路”的仿真，所以分布交互仿真系统有实时性的要求，即要求实体状态必须是实时更新的、实体间的信息必须是实时传播的、图形显示必须是实时生成的等。此外，DIS系统必须保证仿真系统中的时间和空间与现实世界中的时间和空间的一致性。时空一致性是指DIS各节点或各软件对象的行为根据所模拟的时空关系，按严格规

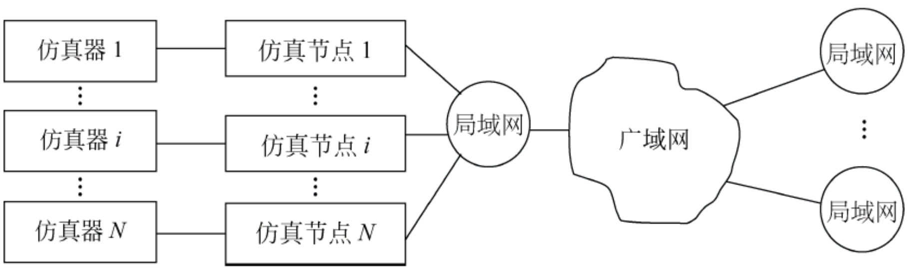


图1 典型的分布交互式仿真系统的结构

定的时序进行,以使用户产生逼真的时空感受。为此必须建立统一的时间和空间基准。时空一致性原则要求 DIS 网上各节点在交互作用时,其状态信息应具有统一的几何和时间表达。

(5) 开放性　开放性是指体系结构的开放性,分布式仿真系统建立一种具有广泛适用性的系统结构框架,在这一框架下可以实现各类系统或子系统的集成。当新的仿真子系统连接到一个分布式仿真系统时,它带来了自己的资源,并实现其资源与其他仿真子系统进行互连、互通、互操作,以构建大规模的和多用途的仿真系统。

参考文献

1. 柏彦奇. 联邦式作战仿真. 长沙:国防科技大学出版社,2001

2. 郭齐胜. 分布交互仿真及其军事应用. 北京:国防工业出版社,2003

3. 肖田元,范文慧. 系统仿真导论. 2 版. 北京:清华大学出版社,2010　　(范文慧　肖田元)

fenbushi caozuo xitong

分布式操作系统(distributed operating system)　用于支持分布式处理系统的**操作系统**。和集中式操作系统相比,分布式操作系统在资源管理、进程通信和系统结构三方面有明显的区别。对于分布式操作系统而言,如果仍采用一类资源由一个管理者来管的集中管理方式,那么,当多个结点的用户要求使用某个结点上的一类资源时,可能需要频繁通信,增加开销,所以分布式操作系统采用一类资源多个管理者的分布式管理方式。多个管理者相互协调,共同完成对资源的管理,以提高管理效率。分布式操作系统的进程通信有两种主要形式:消息传递和远程过程调用(RPC)。消息传递以进程间直接相互发送或接收消息实现,RPC 则通过调用远程进程的过程实现消息的交换。在系统结构方面,分布式操作系统和集中式操作系统一样,也是由内核及提供系统各种功能的模块组成。但在分布式系统中,每台计算机上都有一个内核,它实现对该计算机系统进行基本的控制,系统的其他模块则不均匀地分布在各台计算机上。这种分布可以是静态的,也可以是动态的,它不仅可节省系统开销,而且可以保证系统的稳健性。

分布式操作系统的机制实现可分为两类:面向进程模式和面向对象模式。面向进程的模式是通过一组进程来提供上述服务。通常,这些进程的规模较大,因此一个分布式操作系统中进程数目不多。进程间的同步及对用户进程的控制都是通过消息传递实现的。面向对象的分布式操作系统由一组对象组成,它们提供对系统的各种操作,这样的分布式操作系统由大量对象组成,每个对象负责在自己的局部环境中执行少量的操作。

采用面向对象方法、线程机制和微内核结构及客户-服务器工作模式是当前分布式操作系统研究和设计中的主要研究方向。

参考文献

1. 孙钟秀. 分布式计算机系统. 北京:国防工业出版社,1987

2. Fortier P J. Design of distributed operations systems, concepts and technology. New York: Intertext Pub. and McGram-Hill, 1986　　(杜兴)

fenbushi chengxu sheji

分布式程序设计(distributed programming)　适用于分布式处理系统的一种程序设计。它出现于 20 世纪 70 年代后期。

采用分布式程序设计时,一个程序由分布于系统各结点上的程序模块组成。这些模块可以同时执行,它们通过通信互相协调和交换数据。与经典的程序设计相比,分布式程序设计有三个特点:分布、通信和坚固性。

分布是指程序分为若干个可独立执行的模块,它们分布于系统的不同结点上。程序的分布可分为静态和动态的两种。静态分布是指在装入程序时就将组成程序的模块分别装入不同的结点,在程序执行时不再增加独立执行的模块。动态分布是指在程序执行的过程中产生可独立执行的模块。一个模块在执行中可以撤销它所创建的模块,在它自己被撤销时,一般说,也应将它创建的模块撤销。

通信是指各程序模块之间传递消息。由于分布于各个结点上的程序模块是相互关联的,它们在执行中需要交换数据和互相协调,因此通信是必不可少的。在分布式程序设计中相互通信的两个模块往往是位于两个不同的结点上,通信必须通过连接各个结点的网络进行。这种通信的实现比位于同一台计算机内的两个进程之间的通信的实现要复杂得多。

坚固性是指当系统的某些结点失效时系统仍能维持工作的能力。由于系统是由多个结点组成,从硬件角度看它可能具有坚固的性能。但是要真正具

有坚固性还必须依靠软件。当出现结点失效的故障时系统除作一些信息保护外是无能为力的。具体的维护工作只有用户程序的编制者才知道。因此,分布式程序设计语言往往要给用户提供一些手段来处理结点失效。

采用分布式程序设计来处理问题时必须提供分布式程序设计语言。分布式程序设计语言必须提供程序分布、程序通信和故障处理三种功能。设计一种分布式程序设计语言有两种方式:在某种程序设计语言上增加分布和通信等功能或重新设计一种新的语言。

参考文献

孙钟秀. 分布式计算机系统. 北京: 国防工业出版社,1987

(孙钟秀)

fenbushi chuli xitong

分布式处理系统(distributed processing system) 将不同地点的或具有不同功能的或拥有不同数据的多台计算机用通信网络连接起来,在控制系统的统一管理控制下,协调地完成信息处理任务的**计算机系统**。分布式处理系统是**计算机网络**在功能上的延伸,分布式处理系统中的多台计算机除了可以相互通信和共享资源外,还能协同工作。分布式处理和并行处理的界限对某些系统来讲是模糊的,例如工作站簇既是分布式处理系统,又可看成是松散耦合的并行处理系统。一般认为,集中在同一个机柜内或同一个地点的紧密耦合多处理机系统或**大规模并行处理**系统是**并行处理系统**,而用**局域网**或广域网连接的计算机系统是分布式处理系统。松散耦合并行计算机中的并行操作系统有时也称为分布式操作系统。

分布式处理系统可以有多种组成方式,可由机型和功能大致相当的计算机按水平方式组成,也可由主机系统和**工作站**、**个人计算机**、**终端**等按垂直方式组成。**客户-服务器计算**系统、计算机簇、异构型计算机系统(参见**分布式异构型计算机系统**)、**计算机支持的协同工作**系统等都是分布式处理系统的例子。

分布式处理系统包含硬件、控制系统、接口系统、数据、应用程序和人6个要素。

硬件包括计算机本身的硬件装置,例如处理器、存储器、输入输出设备等,还包括各种网络通信设备,例如各种局域网、广域网、**网络互联设备**、交换机以及各种传输媒体等。

控制系统是分布在各种硬件装置中的控制程序的集合。控制系统主要包括:①**分布式操作系统**,例如Mach操作系统;②分布式数据库(参见**分布式数据库系统**)和分布式资源管理工具,例如Sybase数据库;③通信协议(参见**网络协议**)和网络管理协议,例如OSI协议和**TCP/IP协议集**;④不同媒体的输入输出控制程序等。

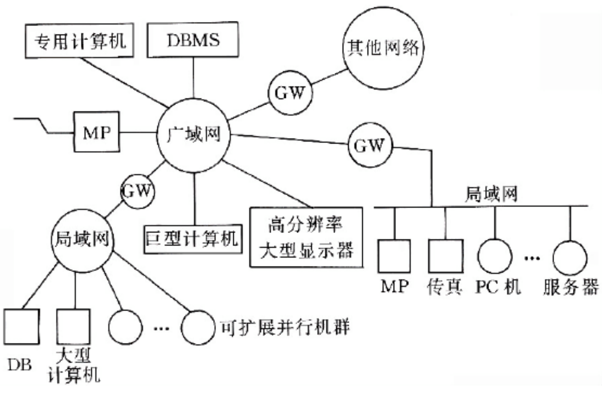
接口系统包括两个部分,即应用程序开发人员接口与一般用户接口,应用程序开发人员接口有时也称为群体开发平台,包括各种语言的库函数和开发工具,还具有对各种不同应用命令的分布协调处理、信息共享与互斥的控制、不同用户和不同媒体的识别以及多媒体信息输入输出处理等功能。一般用户接口包括多种操作命令和图形界面,除了Windows, Motif以及Openlook等窗口系统提供友好的图形界面外,还可提供声音输入、手写体输入以及电视会议系统的实时多媒体输入等的方法。

数据指存储在分布式数据库中的各种数据。

应用程序包括的范围非常广泛,常用的有**文件传送**系统、**电子邮件**系统、电子数据交换系统、计算机集成制造系统等。另外,分布式会议系统、远程教育系统、分布式科学工程计算系统、分布式CASE以及远程医疗监护系统等都是分布式系统的应用例子。

人是分布式处理系统的使用者,设计分布式处理系统时必须考虑使用者所具有的能力和知识,特别是在群体计算系统中,系统设计人员还必须考虑使用者的文化习惯和所处的社会结构等因素。

分布式处理系统的典型物理结构如图1所示。在图中所示的分布式处理系统中,用户通过工作站、



DB —— 数据库; DBMS —— 数据库管理系统;
MP —— 多媒体处理器; GW —— 网关

图1 分布式处理系统的典型物理结构

个人计算机以及终端等透明地使用分布式系统的各种资源。这些资源包括服务器、大型数据库系统和具有大规模计算能力的超级计算机等。用户只需向系统提交任务，系统根据负载情况自动调配各种资源去完成用户提交的任务，用户不必了解系统完成任务的过程和具体细节。分布式处理系统的软件结构如图2。

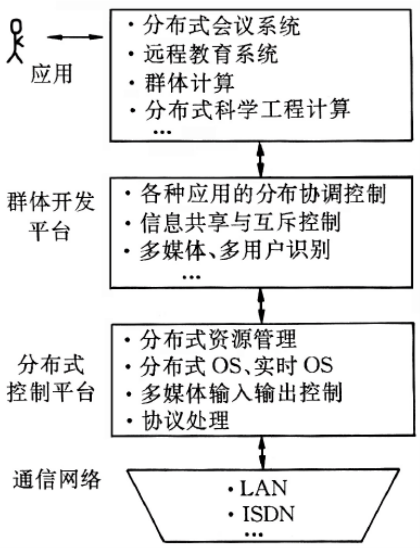


图2　分布式处理系统的软件结构

图3给出了从系统管理人员的角度所看到的分布式处理系统的逻辑结构。用户通过工作站、个人计算机等的控制程序与外部系统连接，而不必知道外部资源的确切位置。

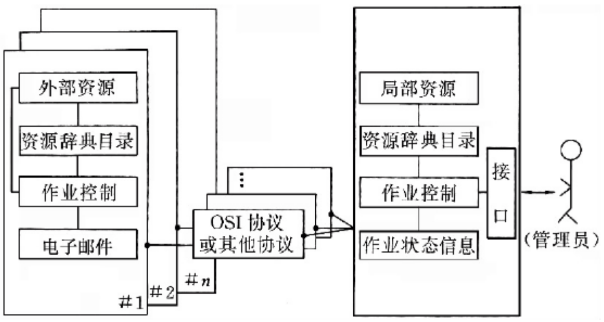


图3　分布式处理系统的逻辑结构

随着网络技术、分布式操作系统和分布式数据库等的发展，分布式处理系统已越来越成熟，从而也得到了越来越广泛的应用。　　（张尧学）

fenbushi cunchu xitong

分布式存储系统（distributed storage system）

数据分散存储在与网络互联的多台独立的设备上，提供并行访问的存储系统。系统中通常采用数据冗余技术提高可靠性，采用负载均衡、锁机制等提高存取效率等。分布式存储系统大体可分为集群存储系统和面向**互联网**的分布式存储系统两种类型。

集群存储系统　主要面向并行计算环境，强调高性能的文件共享和良好的可扩展性，用户不需要考虑文件是存储在集群中什么位置，仅仅需要使用统一的界面就可以访问文件资源。当负载增加时，只需在服务器集群中增加新的服务器就可以提高文件系统的性能。集群存储系统在高性能计算、视频处理、**遥感信息处理**等领域得到了广泛的应用和关注。在网络延迟大的广域网环境下它的体系结构存在一定的局限性，无法胜任面向广域网的大规模数据存储应用。

采用并行文件系统对集群存储系统的数据进行组织管理，底层可以采用基于光纤通道（fiber channel）的**存储区域网**（storage area network，SAN）、**附网存储**（network attached storage，NAS）或者对象存储设备（参见**对象存储系统**）。通过并行文件系统提供标准的 POSIX 接口。

面向互联网的分布式存储系统　Internet 的发展使数据的异地存储成为可能，随着信息数字化的深入，越来越多的数据存储通过网络进行，这些对存储系统提出了更多的需求。在 Internet 中，数据分布的物理距离非常广阔，数据存储的平台种类繁多，为了实现 Internet 范围内数据的访问和共享，人们正努力发展面向 Internet 的海量存储技术，以有效管理分布在广阔范围内和不同平台上的数据。面向互联网的分布式存储系统针对大规模网络服务，从系统的可靠性、文件访问效率等方面对分布式存储系统的架构进行革新。典型的面向互联网服务的分布式存储系统包括：网格（grid）存储、对等存储（即 P2P 存储）和**云存储**。

（1）网格存储　网格将遍布全球的数以万计的计算节点通过高速互联网连接并组织成一个巨大的计算机系统，使其能够透明、高效地完成复杂计算任务。其中，广域网范围的数据管理是网格计算环境需要解决的一个重要问题，人们提出了一些解决方案。例如，在科学计算与高性能计算领域，在现有的文件系统和数据库系统的基础之上，提供一层中间件，利用中间件对这些分布的、异构的系统进行统一管理，使其形成一个整体，共同对外提供数据服务。

（2）P2P 存储　在 P2P 网络环境中，成千上万台彼此连接的存储节点都处于对等的地位，整个网

络一般来讲不依赖于专用集中存储节点。网络中的每一台存储节点既能充当存储服务的请求者,又能对其他存储节点的请求做出响应。

(3) 云存储 与**云计算**类似,它是指通过集群应用、网格技术等,将网络中大量各种不同类型的存储设备通过应用软件集合起来协同工作,共同对外提供数据存储和业务访问功能的一个系统。

参考文献

1. 张江陵,冯丹. 海量信息存储. 北京:科学出版社,2003

2. Shepler S, Callaghan B. RFC 3530: Network File System (NFS) version 4 Protocol. The Internet Society, 2003

3. Foster I, Kesselman C, Tedesco J, et al. GASS: a data movement and access service for wide area computing systems. In: Proceedings of the Sixth workshop on I/O in Parallel and Distributed Systems. May 1999

4. Stoica Ion, Morris R, Karger D, et al. Chord: a scalable peer-to-peer lookup service for internet applications. In: Proceedings of the ACM SIGCOMM'01 Conference. San Diego, California: August 2001

(曾令仿)

fenbushi duomeiti xitong

分布式多媒体系统(distributed multimedia system) 集成了通信、计算和信息处理的分布式系统。它能处理、传送、管理和展现多媒体信息并具有服务质量(QoS)的保证。

分布式多媒体系统的主要特点有:①技术集成:它集成了信息、通信和计算系统,形成一个统一的数字处理环境;②多媒体集成:在这个集成环境中不仅要处理离散的(与时间无关的)媒体数据,还要处理连续的(与时间有关的)媒体数据;③实时性能:要求其存储系统、处理系统和传输系统都具有实时性能,因此,海量存储、高输入输出速率、高网络带宽和高CPU处理能力是必需的;④系统级的服务质量保证:支持沿着从发送者,经传输网络,到接收者的整个数据路径的端到端的不同的服务质量需求;⑤交互性:支持用户和系统之间的双向通信,并允许每个用户都能控制所访问的信息;⑥多媒体同步支持:保持单一连续媒体流内媒体帧的连续播放和相关的数据对象之间的时序关系;⑦支持标准化:在信息内容、展现格式、用户界面、网络协议和信息设备等异构的情况下,能够实现互操作。

分布式多媒体系统的支撑技术分为3个方面。①信息子系统(存储方面):包括多媒体服务器、信息建档、多媒体数据库系统,支持多媒体信息的储存与检索,能一致、安全、可靠地管理数据,响应大量并发的用户请求并保证服务质量。②通信子系统(传输方面):包括传送媒体和传输协议,它将用户与分散的多媒体数据源连接起来,按不同的服务质量传送多媒体数据,如实时传送视频、音频数据,无差错地传送文本数据。随着无线通信技术的进步使得各种便携计算设备,例如笔记本电脑、个人数字助理(PDA)等也能方便地接入系统;③计算子系统(处理方面):包括多媒体计算机、操作系统、展现和著作工具以及多媒体管理软件;用户通过计算子系统管理多媒体数据。

分布式多媒体系统的应用可以划分为3类典型的应用系统:①**流媒体**:在此类系统中,用户可随时访问媒体数据源,例如交互式视频游戏、新闻点播、视频节目点播等;②远程协作:此类系统支持多个用户克服时空阻隔共同参与完成一项任务,例如远程教学、多媒体电子邮件、电视会议、桌面会议、远程医疗、协同编著等;③超媒体:此类系统提供超媒体文档的浏览,例如数字图书馆、电子百科全书、多媒体杂志、多媒体文档、多媒体信息亭、多媒体课件和万维网(WWW)浏览等。 (徐光佑)

fenbushi gongxiang cunchu

分布式共享存储(distributed shared memory) (参见**共享存储**)。

fenbushi jisuan huanjing

分布式计算环境(distributed computing environment) 在具有多地址空间的多**计算机系统**上进行计算和信息处理的软件环境。过去,分布式计算环境是指为分布式计算机系统提供的软件环境。后来,这一概念被扩展为包括各种分布式存储的并行计算机系统提供的计算环境。这些系统的主要特征是多个用户进程在多个通过网络互联的计算机结点上运行,每一进程有自己的地址空间,进程之间通过消息传递模式进行通信。分布式计算环境为这些分布计算提供各种服务和工具,以实现资源共享、并行计算和高可用性。

分布式系统或并行系统可由不同类型的计算机

节点组成，这些节点可实现资源共享。假设一个系统包括几个功能强的节点和数百个功能弱的节点，功能强的节点有强大的计算、存储能力和丰富的软件和信息资源，但价格昂贵；功能弱的节点价格便宜。分布式计算环境能让弱节点上的用户利用强节点上的服务，分享强节点上的各种资源。

通过并行计算技术，分布式计算环境可以集合多个节点上的处理器、内存和硬磁盘资源，用于解决大问题，提高系统吞吐率，减小响应时间。比如某一应用程序需要 2 GB 的内存和 1 年的计算时间，通过数据分割、工作分割等并行技术，此程序可以被分布到 16 个节点上。在理想情况下，每一节点只需有 128 MB的内存，整个程序可在不到一个月的时间内算完。分布式计算环境可为这样的并行应用提供程序设计、通信、任务管理和并行输入输出等的工具。

并行分布式系统应具有很好的可用性，即整个系统不应因为部分节点出故障而崩溃。一个理想的分布式计算环境应该提供一定的故障处理能力，例如故障检测，自动备份，将故障通知系统管理员和用户，重构系统让应用程序绕过故障节点，将程序从故障节点迁移到正常节点等。

下面简述 3 个分布式计算环境，它们是：分布式计算环境（DCE），并行虚拟机（PVM）和消息传递接口（MPI）。

1. 分布式计算环境（DCE）是由开放软件基金会（OSF）于 1992 年发布的分布式计算环境标准，已被广泛采用。DCE 包括如下关键概念和技术。

（1）线程　DCE 支持在同一进程的地址空间内创建和管理多个线程，每个线程是一控制流序列。线程比进程开销小。多线程使一个服务器进程能同时为多个客户服务。

（2）远程过程调用（RPC）　RPC 允许一个节点上的应用程序能调用另一节点上过程，就像调用本地过程那样方便。

（3）名录服务　名录服务存放并管理系统中各种资源的名称和属性等信息，以便任一节点上的任一进程查找。

（4）分布式文件系统　分布式文件系统使用户可以访问任一节点上的文件，就像访问本地节点上的文件那么方便。分布式文件系统也应具有一定的容错功能。

DCE 的其他关键技术还有：安全服务可提供安全通信和安全资源访问，分布式时钟服务可将各节点机的局部时钟同步，无盘服务可使无硬磁盘的节点能使用其他节点的硬磁盘。

2. 并行虚拟机（PVM）是美国的一些大学和国家实验室在 20 世纪 90 年代初开发出来的公开的分布式计算环境软件，已在世界范围内被广泛采用。PVM 通用性强，所需资源较少。将多个计算机用任何网络连接起来，再装上 PVM，即成为一个分布式计算机系统。该系统的节点可用不同的平台，从**微型计算机**到超级计算机均可，因为 PVM 支持异构系统。PVM 提供的服务主要有虚拟机管理、进程管理和**消息传递**。

3. 消息传递接口（MPI）是由欧美的 40 多个国家实验室、大学和计算机厂商共同制订的一个公开的消息传递标准。1994 年发布了第一版，1995 年开发了可移植于多种平台的 MPI——MPICH，1996 年发布了第二版 MPI-2 的草稿。由于其公开性、标准性、可移植性和高性能消息传递功能，MPI 已被广泛采用。MPI 比 PVM 具有更强的消息传递功能。MPI-2 还增加了进程管理、单边通信和并行输入输出等功能。

另外，值得重视的还有**万维网**（WWW），由于采用公开的、标准的、与平台无关的超文本描述语言以及免费的浏览器软件，它提供了方便、友好的界面，使用户通过互连网络能查询和传递各种类型的信息（文本、图像、声音、动画等）。Sun Microsystems 公司于 1995 年推出的 **Java 语言**可让用户通过互连网络执行各种分布式程序。有人认为，今后的主流分布式计算环境将是基于万维网的、与平台无关的窗口系统。（徐志伟）

fenbushi jisuan moxing

分布式计算模型（distributed computing model）　描述分布式软件系统中各软件实体所担当的角色以及与其他软件实体间通信方式的一种计算模型。典型的分布式计算模型包括：

（1）主机-终端模型　一台主机与多个可操作终端直接连接，主要的计算任务由主机程序完成，终端仅实现某些与输入输出有关的辅助任务，用户通过终端操作主机。这一模型是 20 世纪 80 年代以前的主流计算模型，严格意义上说，该模型并未实现计算本身的分布化。

（2）**主/从模型**　在分布式软件系统中，由一个主软件实体单向控制一个或多个其他软件实体。该模型是一种逻辑上紧耦合的模型，以严格的控制与被控制关系为基础，具有结构紧密、协作高效等特

点，曾在早期的分布式操作系统和分布式数据库管理系统中得到应用。但开放性差，难以构建互联网环境下集中控制的分布式应用。

（3）**客户/服务器模型** 把任务分割成不同的部分，在具有前后端结构的分布式计算环境中运行，前端客户机上通常运行应用程序的主控程序和其他辅助性程序，计算量或数据量较大的计算任务一般放在后端的一个或多个服务器上，由客户机程序通过服务请求，从服务器获得其需要的计算结果。该模型逻辑上是松耦合的。目前广泛应用的是在两层客户/服务器模型上发展起来的三层或多层客户-服务器模型，例如“浏览器-Web服务器-应用服务器-数据服务器”模型。

（4）**对等模型** 软件实体处于对等地位，每一个软件实体既能充当软件服务的请求者，又能充当服务的提供者，对等模型具有较高的资源利用率，已在互联网计算环境下广泛用于各类服务共享。

（5）**发布-订阅模型** 由发布者、订阅者和事件服务三类软件实体组成的基于事件的模型，发布者对外发布事件，订阅者事先订阅感兴趣的事件，并在相应事件发生时得到通知。两者通过事件服务进行交互，并实现时间、空间、控制流的解耦，因而通常能够很好地满足分布式系统松散通信的需求。

随着分布式软件系统日益复杂化和服务化，特别是以互联网为代表的广域、自治、异构分布计算环境的出现，多层客户/服务器模型、对等模型和发布/订阅模型将得到越来越广泛的应用。

（吴泉源 丁博）

fenbushi jisuan shineng jishu

分布式计算使能技术（distributed computing enabling techniques） 保障**分布式软件系统**中软件实体之间顺畅通信和高效可靠运行的基础性关键技术。

通信方面的使能技术主要有**套接字**，以及基于套接字的**消息传递**、**远程过程调用**、**对象请求代理**等技术和**互操作协议**；运行方面的使能技术主要有**并发控制**、**服务组合**、**服务容器**和**服务质量保证**等。

（吴泉源 丁博）

fenbushi jujue fuwu

分布式拒绝服务（distributed denial of service） 拒绝服务攻击的一种主流形式。拒绝服务（denial of service）是阻止合法用户使用网络服务的攻击技术，造成网络服务可用性的破坏。分布式拒绝服务攻击，简称DDoS，通过利用大量计算机向目标服务发送数据，造成目标服务某种关键资源的耗尽，从而使其不能为合法用户提供可用服务。

DDoS攻击可以分为两大类，即直接攻击与反射攻击。

在直接攻击下，攻击者向目标主机发送大量数据包，造成目标主机系统资源或者网络带宽耗尽而无法提供正常服务。根据数据包类型的不同，又分为SYN洪泛攻击（SYN flooding）、UDP洪泛攻击（UDP flooding）、ICMP洪泛攻击（ICMP flooding）和应用层洪泛攻击等。其中SYN洪泛攻击是最常见的攻击技术，这种攻击利用TCP协议的三次握手机制中的缺陷，通常结合IP网络地址欺骗技术伪造大量SYN包发往目标服务器，使得目标服务器不得不维护非常大的半开连接列表，耗尽系统资源造成拒绝服务。

反射攻击是一种间接攻击方式，又称为分布反射式拒绝服务（DrDoS），攻击者利用某些网络协议的放大效应（广播应答），将请求数据包的IP源地址设置为目标主机的地址，向中间的反射节点大量发送，而反射节点则将流量更大的响应数据包发往被攻击的目标主机，造成目标主机网络被淹没。分布反射式拒绝服务具体攻击技术包括ICMP Smurf攻击、DNS响应洪泛攻击等。

DDoS攻击最早出现于20世纪90年代末期，2000年2月份接连出现的针对Yahoo!、eBay等网站的DDoS攻击使其被业界所高度关注和重视。黑客社区也出现了一些专用DDoS攻击工具，著名的有Trinoo、TFN、TFN2K、stacheldraht等。随后僵尸网络的快速发展为DDoS提供了更加强大的攻击平台。

DDoS攻击是互联网最难解决的安全威胁之一，其产生根源在于互联网本身的开放性以及底层协议缺乏验证机制。目前的DDoS防范技术措施有：①避免终端主机成为傀儡与僵尸主机，从而预防DDoS攻击；②通过IP真实地址验证技术避免地址欺骗，并使用IP追踪技术追查攻击者真实的源地址；③采用防火墙、黑洞与流量清洗等技术来进行DDoS攻击过滤与保护。

DDoS已成为互联网上一种普遍存在的安全威胁，并将长期存在。需要从下一代互联网体系结构和安全模型开展深入研究，才能够彻底消除DDoS

攻击威胁。

参考文献

1. 徐恪, 徐明伟, 吴建平. 分布式拒绝服务攻击研究综述. 小型微型计算机系统, 2004, 25(3): 337-346

2. Mirkovic J, et al. A taxonomy of DDoS attacks and DDoS defense mechanisms. ACM SIGCOMM Computer Communication Review, 2004, 34(2): 39-53

（诸葛建伟）

fenbushi ruanjian xitong

分布式软件系统(distributed software system) 支持分布式处理的软件系统。主要有**分布式操作系统**、**分布式数据库系统**、**分布计算中间件**和分布式应用软件系统等。

分布式操作系统负责管理分布式处理系统的资源和控制分布式程序的运行。它在资源管理、进程通信和系统结构等方面与集中式操作系统有明显区别。

分布式数据库系统由分布于多个计算机节点上的若干逻辑上相关的子数据库系统组成,它提供有效的存取手段来操纵这些节点上的子数据库,使之在使用上可视为一个完整的数据库。

分布计算中间件是在网络环境中,建立在操作系统之上,支持分布式应用软件开发、部署、运行和管理的软件系统,如消息中间件、事务处理中间件、面向对象中间件、应用服务器、企业应用集成中间件等。分布计算中间件可以看作紧耦合的传统分布式操作系统在网络开放环境下的发展,它所涉及的网络各节点操作系统和各类网络资源通常都具有明显的自治性,甚至是异构的,它们之间不仅地理上分布,逻辑上也是松散耦合的,其联系主要靠互操作协议实现。

分布式应用软件系统是运行在网络中多个计算节点上的应用软件系统,如网络环境中的各种**管理信息系统**、**电子商务系统**、**电子政务系统**、飞机订票系统,即时通信系统,网上购物系统等。

分布式软件系统的发展已有几十年历史。20世纪70年代中期到80年代初期,其主要目标是发展分布式系统软件来支持实验性的分布式硬件系统,以成为一个分布式处理系统,出现了支持通信和资源共享的网络操作系统和分布式操作系统,但都未达到实用。80年代中后期开发了一些具有一定实用价值的分布式文件系统、分布式操作系统,以及远程过程调用通信机制等。20世纪90年代以来,随着网络技术的快速发展及网络应用的不断深入,分布式软件系统进入实用化阶段。鉴于对具有统一控制、逻辑上紧耦合,而忽视网络应用系统固有的自治性、异构性和可演化性的反思,松耦合的分布式软件系统成为主流得到了迅速发展,分布式计算环境、**分布式计算模型**、**分布式计算使能技术**和分布计算中间件的研究和产业化取得了突破性进展。目前,中间件已成为快速构建分布式应用软件系统的重要平台。

随着新一代互联网和云计算、物联网等信息网络技术的发展,分布式软件系统将朝着构件化、服务化、泛在化的方向不断发展。

参考文献

1. Ananda A L, Srinivasan B. Distributed computing systems: concepts & structures. IEEE Computer Society Press, 1991

2. An Overview of distributed Applications visual studio. Net 2003, MSDN library

（吴泉源　谢立）

fenbushi shujuku

分布式数据库(distributed database) 一组逻辑相关的分布在计算机网络上的多个数据库的组合。**分布式数据库管理系统**(distributed database management system,DDBMS)是管理分布式数据库并使之对用户分布透明的软件系统;分布式数据库和管理它的 DDBMS 一起被称为**分布式数据库系统**(DDB System,DDBS),有时简称为分布式数据库。

网络中每个站点逻辑上由单个计算机组成,它们之中不存在一个集中控制。位于站点上的数据库被称为**局部数据库**(local database)或**组件数据库**(component database)。从整体上看分布式数据库也是一个数据库,相应的称它为**全局数据库**(global database)。分布式数据库有同构和异构之分,如果所有的局部数据库具有相同的数据模型,则称为**同构型分布式数据库**(homogeneous distributed database)。第一代分布式数据库系统(又称传统分布式数据库系统)主要研究同构型分布式数据库,局部数据库基本上都是基于关系模型的。在**异构型分布式数据库**(heterogeneous distributed database)中,不同的局部数据库或组件数据库可能具有不同的体系结构、数据模型、数据模式、用户接口语言和事务处理策略等。新一代分布式数据库系统主要研究异构型分布式数据库。异构型分布式数据库系统的目的是使用

户能透明访问系统中不同类型的组件数据库。设计一个异构型分布式数据库的主要策略是封装组件数据库的异构性并尽可能采用已有的同构型分布式数据库技术。

分布式数据库系统在集中式数据库系统的组成基础上增加了三个部分:DDBMS、分布式目录和网络访问进程。分布式目录为DDBMS提供了数据定位的必要信息,网络访问进程使用高级协议来执行局部站点和分布式数据库之间的通信。分布式数据库系统具有本地自治性和分布透明性等重要特性。**本地自治性**(local autonomy)是指局部数据库系统可以自己决定本地数据库的设计、使用以及与其他节点的数据库系统的通信。**分布透明性**(distributed transparency)是指分布式数据库管理系统将数据的分布封装起来,用户访问分布式数据库好像在自己的计算机上与集中式数据库打交道一样,不知道也不必关心数据的存放和操作位置等细节。

四层结构是分布式数据库系统的一种结构框架,它由全局外层、全局概念层、局部概念层和局部内层组成。全局外层是最接近用户的层次,由多个全局用户视图组成,它是分布式数据库的全局视图;全局概念层描述分布式数据库系统中全局数据的逻辑结构和数据特性;局部概念层是局部数据库的概念视图,每个局部数据库还可以提供局部外模式以支持局部数据库的应用;局部内层是局部数据库的内部视图或存储视图,它与数据库的实际存储密切相关。

这四层分别对应于全局数据库的外模式和概念模式以及局部数据库的概念模式和内模式。分布式数据库系统的分层对于保证数据的独立性和分布透明性具有特别重要的意义。

在四层结构间存在着三级映射:全局外层-全局概念层映射定义了全局用户视图与全局概念视图之间的对应,使分布式数据库具有数据的全局逻辑独立性;全局概念层-局部概念层映射定义了分布式数据库的全局概念视图与局部数据库的概念视图之间的对应,保证了分布透明性;局部概念层-局部内层映射定义了局部数据库的概念视图与存储的数据库之间的对应,保证了数据的物理独立性。为了给数据的分布透明性提供方便,全局概念层引入了数据的分片(fragmentation)和分布模式。数据分片的主要功能是把一个逻辑上完整的数据库划分成若干部分。在分片设计中,每个全局关系被分成若干个不相重叠的部分,称为数据片。分片的结果要使得全局关系的所有数据必须映射到某个数据片中,并且分片后得到的所有数据片可重构原来的全局关系。

分布式数据库系统是在集中式数据库系统和计算机网络的基础上发展起来的,它是分布式数据处理的关键技术之一,在办公自动化、管理信息系统等领域有着广泛的应用。分布式数据库系统从20世纪70年代诞生以来取得了长足的进展,20世纪80年代是分布式数据库系统研制的一个高峰时期,到20世纪80年代末,第一代分布式数据库系统的理论已经成熟并已有产品问世。具有代表性的第一代分布式数据库系统有SDD-1、POREL、R※、分布式INGRES系统、SIRIUS计划和ADA-DDM系统等。近年来,Internet的普及和大规模异构数据的应用使分布式数据库技术的研究有了飞跃的发展,新一代分布式数据库系统正日趋成熟。

参考文献

1. 李昭原,等. 数据库技术新进展. 2版. 北京:清华大学出版社,2007

2. Oezsu M T, ValDuriez P. Principles of distributed database systems 2nd ed. Prentice Hall, 1999

(周龙骧 王翰虎 郑宇华)

fenbushi shujuku xitong

分布式数据库系统(distributed database system) 由分布于若干个计算机结点上的若干子数据库系统所组成的**数据库系统**,它提供有效的存取手段来操纵这些子数据库。**分布式数据库**在使用上可视为一个完整的数据库,而实际它是分布在地理上分散的各个节点上。分布在各个节点上的子数据库在逻辑上是相关的。操纵这些子数据库的软件称为分布式数据库管理系统(DDBMS)。

分布式数据库系统在集中式数据库系统的组成基础上增加了三个部分:分布式数据库管理系统DDBMS(又称网络数据库管理系统NDBMS),网络数据字典NDD和网络存取进程NAP。

分布式数据库管理系统的主要功能如下:

(1) 接收用户请求,并判定送往何处,或必须访问哪些结点计算机才能满足该请求。

(2) 访问网络数据字典,或者至少了解如何请求和使用其中的信息。

(3) 如果目标数据存在于系统的多个结点计算机上,就可以进行并行处理。

(4) 在用户进程、局部数据库管理系统和其他计算机的数据库管理系统之间进行通信服务。

(5) 在一个异构型分布式处理环境中,还需要提供数据和进程移植的支持。所谓异构型是指在系统的每个结点上的数据库系统是有差别的。

网络数据字典为分布式数据库管理系统提供达到其功能的数据单元位置的必要信息。网络存取进程提供一个结点的分布式数据库管理系统的进程和通信子系统之间的接口。它使用高级协议来影响内部节点的分布式数据库之间的通信。

从结构上分,分布式数据库系统可分为两类:**全局型分布式数据库系统**和**联邦型分布式数据库系统**。

全局型分布式数据库系统是最常用的类型。在这种结构中,一个数据库以完整的方式分布在一个计算机网络的各个节点上。节点间的所有信息交换都由系统软件处理,它对用户是透明的。这样结构具有如下特征:采用完全分布方法;允许有多份副本;由分布式数据库管理系统来控制进程间的同步;系统进行故障恢复处理;系统具有较高性能。

采用这样结构的分布式数据库系统中,用户看到的是一个完整的数据库,他们不必了解他们的数据所在位置。同样,对于应用程序而言,通信将在数据库管理系统中进行,并且由分布式数据库管理系统来处理。

联邦型分布式数据库系统的结构用于以下情况:存在若干个现存的、分离的数据库节点,且这些数据库的类型可以相同,也可以不同。这类结构具有如下特征:分离的数据库必须在逻辑上以某种方式联成一个整体;从单个终端用户的观点来看,他们要求访问若干不同的数据库,但是,不必了解新的接口进程。用户所需要的是一个“友善的”和与所有查询相容的接口。同时,事务处理由系统软件自动进行。

联邦型结构往往在“更新”情况下采用,因此受到原有软件的限制。另外,由于开销原因,也不宜有网络范围内的“恢复”机制。最后,尽管有 NDD 和合作的数据字典提供有用的帮助,但用户通常仍需知道它们数据的存放位置。

分布式数据库系统是在集中式数据库系统和计算机网络的基础上发展起来的,它是分布式数据处理的关键技术之一,在办公自动化、管理信息系统等方面有着广泛的应用。例如,银行管理系统,企业管理系统,军事指挥训练系统,航空公司或旅馆的预订系统等。

参考文献

谢立,孙钟秀. 分布式数据处理. 北京:国防工业出版社,1990

(郑宇华)

fenbushi xitong jiance

分布式系统监测(distributed system monitoring) 一种对分布式系统内部运行状态和运行环境信息进行收集、汇总与分析的机制。分布式系统监测可用于分布式系统的系统维护、系统容错、系统评估、软件调试、软件调优和系统管理等,是保障分布式系统正确、可靠运行的一种有效机制。

分布式系统监测机制通常由监测探针、监测代理和监测中心三类监测程序组成。监测探针指埋藏在分布式系统软件实体内部、用于获取其内部运行状态和外部环境信息的一个程序;监测代理是运行于被监测软件所在节点的程序,负责收集运行于本节点各监测探针返回的数据;监测中心负责管理系统中监测代理的运行状态和监测策略,从监测代理收集监测数据,并对监测数据进行汇总和分析。

在分布式系统监测机制中,可以根据实际需要灵活配置和部署上述三类监测程序。对于较简单的系统,监测探针和监测代理可以合并在同一个程序中;对于规模较大的系统,监测代理和监测中心可以设置成多层次的。 (吴泉源　尹刚)

fenbushi xuni huanjing

分布式虚拟环境(distributed virtual environment,DVE) 位于不同地点的**虚拟现实**装置或计算机通过网络连接起来,共同构造的一个可共享虚拟环境的分布式系统。绝大多数分布式虚拟环境都是通过网络连接,又称分布式虚拟现实(distributed virtual reality,DVR)、网络虚拟环境(networked virtual environment,NVE)、多用户虚拟环境(multiuser virtual environment)、协同虚拟环境(collaborative virtual environment,CVE)、共享虚拟环境(shared virtual environment)等。分布式虚拟环境起源于军事仿真演练的需要,是随着计算机网络技术的快速发展和普及而发展起来的,它进一步拓展了虚拟现实研究与应用的范围,在经济建设、国防安全和文化教育等领域有着重大的应用前景。

分布式虚拟环境有基于互联网和基于专用网两大类。基于互联网的分布式虚拟环境可以追溯到20世纪70年代末出现的多用户游戏(multiuser dun-

geon,MUD),也称为基于文本的网络化虚拟环境。1994 年,虚拟现实建模语言(virtual reality modeling language,VRML)首次在 WWW 大会上被提出,并在 1997 年 12 月正式发布为 ISO 国际标准 VRML97(ISO/IEC 14772-1:1997)。VRML 定义了一种支持对交互式三维矢量图形进行描述的文本格式标准,专门面向互联网上的虚拟现实应用。1999 年初,VRML 组织更名为 Web3D 组织,开始制订新的国际标准 X3D(Extensible 3D),X3D 整合正在发展的 XML、JAVA、流技术等先进技术,在 2004 年被 ISO 组织确定为 ISO/IEC 19775/19776/19777 标准。从 VRML 到 X3D 标准,都保留了专门支持分布式虚拟环境的模块,称为 DIS(distributed interactive simulation) component(标准第一部分 28 节)。HTML 5 的 Canvas 元素支持使用不同的渲染引擎来编写图形程序,其中 2011 年 2 月发布的 WebGL 1.0 是一种三维绘图标准,无须插件而三维渲染效果良好,在移动互联网兴起的年代已得到很多厂商支持。

从 20 世纪 90 年代初开始,具有三维图形界面的网络游戏开始出现,开启了互联网上的三维网络游戏时代。大约 1997 年开始,大型多人在线游戏(massively multiplayer online game, MMPOG)迅速崛起,分布式虚拟环境技术在娱乐业得到了大范围应用。2003 年,美国 Linden 实验室发布了 Second Life(第二人生),以游戏的形式提供了大型三维模拟现实世界的分布式虚拟环境。

基于专用网的分布式虚拟环境应用最早、最广泛的是在仿真领域,又称为**分布交互式仿真** DIS(distributed interactive simulation)或**分布式仿真**(distributed simulation)。1989 年形成了世界上第一个分布式虚拟战场环境。1993 年形成了分布交互仿真的 IEEE 1278 标准集。在 DIS 的基础上,美国国防部建模与仿真办公室(Defense Modeling and Simulation Office,DMSO)主持开发了**高层体系结构** HLA(high level architecture),HLA 旨在建立一个通用的高层仿真体系,达到各种模型和仿真的互操作和可重用性。2000 年 9 月,HLA 被采纳为 IEEE P1516 标准,2003 年、2007 年标准两次补充,2010 年进行了重新修订。

随着网络基础设施的发展,基于互联网和基于专用网这两大类的技术出现融合的趋势。

1. 问题抽象

将分布式虚拟环境问题进行抽象,多个用户共享一个虚拟环境,并进行操作,得到相应的反馈,真实物体可以通过仿真或机械等手段进行虚实的交互操作,虚拟场景可以从真实世界的数据生成,如图 1 所示。对这些抽象元素进行应用分析,实现分布式虚拟环境有三个基本目标:①一致性　所有节点能够得到状态一致的虚拟世界,不一致的虚拟世界会导致用户交互的各种问题,一致性是分布式虚拟环境的重要特点和要求;②交互性　交互性是虚拟现实真实感的重要特征,用户之间通过虚拟环境进行交互,并有所反馈,才能有身临其境的感觉;③实时性　由于人的感知具有时间要求,过长的反馈会影响真实感,虚拟世界中的相互作用和三维绘制需要在人可以容忍的响应时间内完成。

2. 基本概念与技术难点

分布式虚拟环境中的自然环境包括地形、构造物和天空等,用户对它们的交互性很弱,因此更改要求相对较小。人物、物体和计算机生成的角色是虚

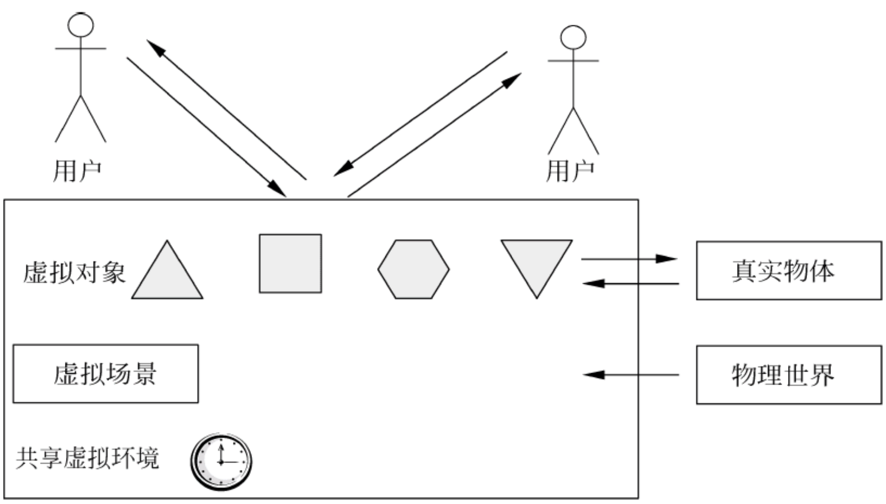


图 1　分布式虚拟环境的问题抽象

拟世界中主要的数据变化主体,统称为对象。从状态的角度看,自然环境和对象的几何模型或应用逻辑可以作为固定的数据或规则在虚拟环境系统运行前确定,应用中一般不需考虑其动态变化。对象的共享则可以通过各节点赋以对象模型相同的空间位置、方向、颜色等参数实现一致性视图,这些参数的集合就是该对象类的状态集,也就是描述对象的量化指标。考虑到对象存在状态共享上的差别,对象还被进一步划分为静态对象和动态对象。静态对象主要指树木、桥梁、房屋等自然环境有关的对象,这些对象一般不会主动移动或发生状态变化,也被称为"被动实体"。动态对象是虚拟世界中主要的数据变化主体。

虚拟环境中的动态变化除了对象的状态外,还会发生一些事件,事件是不依赖于对象的,因此不能作为对象的状态之一,如物体间的碰撞就是一个事件,常见的事件还有开火、爆炸、喊话等。在虚拟环境研究中,将事件的发生称为交互。交互和对象状态都是虚拟环境中动态发生的信息,但交互的比例相对较小。

分布式虚拟环境中的技术难点主要分布在以下几个方向:

(1) DVE 系统结构　经典的集中式、分布式 DVE 模型已经成为主流技术,大规模虚拟环境成为新的要求。多服务器系统、服务器集群、层次式结构模型及其核心算法是当前有关 DVE 系统结构的研究热点。

(2) 动态共享状态　对动态共享状态方法的传统研究主要集中在状态的预测与推算算法上,如 DR (dead reckoning) 算法和延迟补偿。复杂对象的状态共享不能够简单地用少量参数来描述,是研究的难点问题,如动态地形和可变形物体的网络共享技术。

(3) 时空一致性问题　由于网络延迟是不可预测的,分布式虚拟环境中的事件一致性一直是核心难点问题之一,也常被称为时空一致性问题。时间同步算法是解决事件一致性的主要技术,还存在一些技术通过因果序或其他逻辑关系计算实现事件子集的排序。

(4) 系统可扩展性技术　通过软、硬件提高 DVE 系统的可扩展性一直是各种系统中主要研究的技术,如兴趣技术、异类系统的互联技术、编解码算法、包聚合算法、负载平衡算法、多处理器并行优化方法等。其中兴趣技术利用了数据分布的局部性原理,是分布式虚拟环境研究的热点问题。

(5) 其他还有数据可靠性问题、标准制定和应用系统问题等。

随着网络带宽和处理器能力的大幅度提高,在互联网上开展分布式虚拟环境研究和应用成为趋势。目前产业界和国际标准制定以在网页上实现三维交互式图形为热点,如 X3D、HTML5 等,同时,越来越多的研究者将实时采集的数据合成到虚拟环境中进行共享和交互,形成虚实混合的分布式虚拟环境。不同技术的融合和普及化将成为主流。

参考文献

1. 周忠,吴威. 分布式虚拟环境. 北京: 科学出版社,2009

2. 赵沁平. 虚拟现实综述. 中国科学(F 辑: 信息科学), 2009, 39(1): 2-46

（周忠）

fenbushi yigouxing jisuanji xitong

分布式异构型计算机系统(distributed heterogeneous computer system)　由多个不同种类的计算平台或应用子系统通过网络连接而成的**计算机系统**。计算平台(简称**平台**)是指计算机的硬件系统和操作系统的组合。应用子系统是指在平台之上支持某一特定应用的软件,例如数据库。

下面通过几个例子来说明异构性。例如,某一分布式系统由 3 台**微型计算机**通过**以太网**连接而成。所有微型计算机都采用奔腾 Pro 处理器芯片和与 IBM 微型计算机兼容的硬件体系结构,但用了 3 种不同的操作系统: 一台用 Windows 95, 一台用 Windows NT,一台用 Linux。这就是一个异构系统,因为有 3 种不同的平台。再例如,某一分布式系统在它的所有计算机中都采用同一操作系统 Windows NT,但计算机采用了不同的处理器:奔腾和 Power PC,这也是一个异构系统,因为用了两种不同的平台。一个分布式系统的所有平台即使都相同,但如果采用了不同的数据库(例如 Oracle 和 Sybase),该系统也是异构的。

绝大多数分布式计算机系统都是异构的。主要原因有 3 个:第一,用户对计算机系统的需求是异构的。比如用户需要有高性能的计算能力、高输入输出吞吐量的文件服务器和快速的三维图形终端,以及强有力的应用软件。满足这些需求只能采用不同的平台。第二,异构性可以提高系统的性能价格比。例如某一系统可由 1 台昂贵而性能高的服务器和 10 台便宜的微型计算机组成。微型计算机用户可

以通过网络共享服务器的资源。这个异构系统比由11台服务器组成的同构系统价格低廉得多。第三,用户经常依靠连接多个子系统的办法来扩大自身的系统,这样也常常导致异构系统。

异构系统所要解决的最关键的问题是**互操作性**,即允许在多种平台间的数据通信和程序执行。互操作性可用3种技术来实现:第一是采用标准的接口,例如**Internet**上的几百万台计算机都可以用**TCP/IP协议集**通信。第二是用**中间件**,即用来支持平台间相互作用的软件,它支持平台间的数据通信和程序执行。第三是用与平台无关的编程工具,它开发出来的程序可以在不同平台上执行。这样的编程工具包括Java语言。Java的字符数据类型采用16位的国际标准码,包括世界上很多语言。而C语言采用8位的ASCII码,只适用于西文,不足以表述有5万多个字符的中文。Java语言不允许直接调用操作系统的过程,必须通过所规定的接口,这样保证了与平台的无关性。 (徐志伟)

fenceng cunchuqi tixi jiegou

分层存储器体系结构(hierarchical storage architecture) 由具有差别的存储介质构成的不同存储层级。它从离处理器最近的、容量小的、存取速度非常快的存储器开始,到远离处理器的、大容量的、速度慢的存储器结束。这种存储介质上的差别主要是存取速度及存储容量的不同。存取速度快的介质通常都是存储单位成本高,而且容量相对来讲比较低。相应地,存取速度慢的介质通常是为了满足容量与成本方面的要求,即在相同成本下可以得到更大容量。将数据存储在不同层级的介质中,并在不同介质之间进行自动的数据迁移、数据复制等操作,目标是在相同或较低成本下,同时满足性能和容量的两方面需要。几乎所有现代计算机系统都采用分层存储器体系结构。

一般来说,分层存储中,将存取速度最快的那一层称为第0层,依次为第1层,第2层,等等。理论上说,层级的划分可以有很多层。但是,过多的层级会增加数据及介质管理的难度及可用性。通常层级的设定在2~4层,最多在5层左右。

分层存储与计算机的发明与发展相伴相生,已经在计算机存储领域应用多年。在冯·诺依曼提出计算机的“存储程序”模型时就已经包含了分层存储的概念。“存储程序”原理,是将根据特定问题编写的程序从外存储器**辅助存储器**读入到计算机内存储器**主存储器**中,然后按照程序的规定顺序执行指令,直至程序结束。这里的外存储器与内存储器,就是一个分层存储体系结构的最初模型。

后来,分层存储这一概念被EMC等厂商应用在使用不同存储介质的外存储内部,发展为信息生命周期管理(ILM)技术。ILM因为操作复杂、成本较高,实现起来并不理想。近来自动分层存储技术开始出现,使ILM得以方便实现。分层存储发展至自动分层存储,主要是抛弃了识别数据和迁移数据的人工操作,而且,这种数据迁移操作是双向而非单向的迁移,这也是自动分层存储与传统分层存储的重要差别之一。

自动分层存储是通过策略或者数据的访问频率,动态进行数据迁移,使频繁访问的数据存储在速度较快、成本较高的存储介质上,而访问频率较低的数据则被迁移到速度较慢、成本较低的存储介质上,从而提升高速磁盘区的性能、降低系统成本,而且实现大量非活动数据的实时在线。

硬件结构

典型的分层存储器体系结构中各层存储器的特点如下:

第0层是**中央处理器**内部的寄存器。向这个寄存器存取数据的速度非常快,因为它就在处理器芯片上。但是,其容量受芯片面积的限制。

第1层是**高速缓冲存储器**(cache),在处理器和主存储器(又称内存储器)之间,它是小容量、速度快的存储器。cache中存储处理器最需要的数据块;cache多采用SRAM,价格较高。

第2层是**主存储器**(memory),用来存储程序和数据,并作为输入源和输出目的地。主存储器比高速缓冲存储器(cache)大得多,采用动态随机存储器(DRAM)器件。DRAM比静态随机存储器(SRAM)速度慢,但价格低。

第3层是大容量存储器。现在普遍使用大容量**磁盘存储器**,用来实现**虚拟存储器**技术,使处理器识别的主存储器的容量比实际的大得多。用虚拟存储技术,将地址空间划分为页面,页面比块大得多。在需要的时候,页面被调到主存储器内,不需要的时候就送回磁盘。因为机械性寻址的磁盘存储器要比主存储器慢得多,虚拟存储器比主存储器要慢。磁盘存储器容量大,单位成本低。随着外存储层次的自动分层存储技术的发展,可以采取SSD **固态盘**-FC磁盘-SCSI磁盘-SATA磁盘四层存储结构,也可以采取SSD固态盘-FC磁盘-SCSI磁盘-SATA磁盘-磁带

五层存储结构,具体采用哪些存储级别需要根据具体应用而定。

第 4 层是离线存储器,主要作为归档用途,存储短期内不用的数据。通常由**光存储器**、U 盘存储器、**磁带存储器**组成。

发展趋势

在分层存储技术研究中有两个明显的趋势。第一个是,在使用分层存储资源的时候,要区别使用数据预取空间和数据缓存空间,目标是避免缓存污染,提高上层存储空间的效率,最终提升整个计算机系统的输入输出(I/O)性能。在使用多层存储体系结构时,多个层次都有数据缓存和数据预取,这些层次之间的数据缓存系统之间并不清楚对方都缓存了哪些数据,最终导致各个层次缓存数据的高度重复,第一层缓存过的数据,第二层还是缓存,导致第二层缓存的数据不被利用,浪费了宝贵的上层存储空间,丧失了性能改进的空间。

分层存储技术研究中的另一个趋势是,研究如何避免同样的数据在多个层次之间反复缓存,使得不同上层存储空间缓存不同的数据,从而达到上层存储空间充分利用。

参考文献

张广艳. 相变存储器. 中国计算机学会通讯, 2010,6(10):54-58

(张广艳)

fenji cunchu

分级存储(hierarchical storage) 根据数据的重要性、访问频率、性能需求等指标,采取不同的存储方式将数据分别存储在不同存储设备并实现数据自动迁移、获得高的总体性价比的存储技术。

在实际应用环境中,从存储设备的响应模式划分,存储系统通常由在线存储、近线存储和离线存储三层存储组成。在每层中,特别是在线存储层,还可以根据存储设备的性能差异进一步划分为更细的层次。

在线存储 又称工作级的存储,存储设备和所存储的数据时刻保持"立即响应"状态,是可随意读取的,可满足计算平台对数据访问的速度要求。一般在线存储设备为磁盘和磁盘阵列等设备,价格相对昂贵,但性能最好。近年来随着半导体固态存储器的应用日益广泛,在线存储根据存储设备的性能由高到低、存储容量由大到小还可进一步分为固态存储层、磁盘存储层以及磁带光盘存储层等。

近线存储 指将那些并不是经常用到,或者说数据的访问量并不大的数据存放在性能较低的存储设备上。对这些设备的要求是寻址迅速、传输率高。因此,近线存储对性能要求相对来说并不高,但由于不常用的数据要占总数据量的大多数,这也就意味着近线存储设备首先要保证的是容量。

离线存储 主要用于对在线存储的数据进行备份,以防范可能发生的数据灾难,因此又称备份级的存储。离线海量存储的典型产品就是磁带或**磁带库**,价格相对低廉。通常离线存储介质上的数据在读写时是顺序进行的,因此,存储访问速度慢。

参考文献

时成阁. 网络存储导论. 北京:清华大学出版社,2007

(陈俭喜 曹强)

fenshi chuli

分时处理(time-sharing processing) 多个用户可同时与计算机系统交互处理作业的方式。

提供分时处理功能的操作系统称**分时操作系统**,它支持多个用户在各自的终端同时使用一台计算机而每个用户都感到好像自己各有一台独享的计算机。

分时是指把处理器的时间分成**时间片**,按时间片轮流的方式把处理器分派给各联机作业使用。如果一个作业在分得的时间片内不能完成计算,则暂时中断它,把处理器让给另一个作业使用,等待下轮再继续运行。由于现代计算机运行速度很快,用户作业运行轮转也就很快,实现了多用户共享一台计算机算题的方式。

分时处理的主要特点是:

(1) 同时性 若干个终端用户可同时使用计算机;

(2) 独立性 用户彼此独立,互不干扰;

(3) 及时性 用户的请求能在满意的时间内得到响应;

(4) 交互性 用户能进行人机对话,交互方式工作,联机地控制程序。

参考文献

孙钟秀,等. 操作系统教程. 北京:高等教育出版社,1989

(郑宇华)

fenxixing shuju guanli

分析型数据管理(analytical data management) 面向分析型应用的数据管理技术。

与事务型数据管理主要负责数据更新任务相

对，分析型数据管理技术主要面向数据上的分析处理任务。分析型数据管理技术从存储和计算模型的角度来看可以分为基于 SQL 的技术和 NoSQL（not only SQL）模式的技术。基于 SQL 的结构化数据上的分析型数据管理主要采用列存储技术来提高分析型应用的性能。列存储技术针对分析型应用通常只对模式中较少的属性进行聚集计算操作这一特性，采用各属性列在磁盘上独立按列存储的方法，提高分析型应用中的磁盘 I/O 效率，提高列存储的数据压缩效率。同时，列存储缩小了数据宽度，也能有效地提高 CPU 在数据处理时缓存的命中率，从而提高 CPU 的处理效率。列存储分析型数据库在查询处理层依然采用 SQL 标准，在查询计划执行阶段分为两种类型。一种是完全以列代数来替代关系代数完成 SQL 操作，采用“一次一列”和后物化技术的列代数查询处理引擎，能够充分优化列存储在查询处理层次上的 CPU 性能，其典型代表是 MonetDB。另一种是在内存缓冲区动态地将列数据转换为行数据，在查询计划执行阶段采用行存储查询处理引擎完成 SQL 操作，其典型代表是 Greenplum、Vertica、InforBright 等。列存储分析型数据库的数据更新技术主要分为两种，一种是采用以列存储模型为基础的更新事务处理技术；另一种是在列存储系统之上建立一个较小的行存储更新处理系统来负责更新事务处理，并将更新的记录移动到列处理分析系统之中。

非结构化分析型数据管理技术的发展演化出 NoSQL DBMS 的概念。在互联网时代，超大规模和高并发的 Web 2.0 动态网站成为应用的代表，而这类 Web 应用通常将关系数据库事务的强一致性放松为弱一致性（最终一致性）。在模式设计上通常采用简单模式甚至无模式方式，放松对存储模式的要求，减少了大表关联查询。非结构化分析型数据管理的关键问题是海量数据存储、高扩展性和高可用性分布式计算和高并发读写性能等。其中典型的技术体系是 Google 的 MapReduce 技术、GFS 和 Bigtable 技术。MapReduce 是工作在超大规模分布式环境下的一种高可扩展性的算法，其核心思想是通过指定的 Map（映射）函数把一组 key-value 偶对映射成一组新的 key-value 偶对，指定并发的 Reduce（化简）函数，用来保证所有映射的 key-value 偶对按共享键组划分，并分发到对应的 Reduce 节点上进行处理。GFS 是 Google File System 的缩写，是一种大型廉价集群上处理海量数据的分布式文件系统。GFS 以 Google 业务系统的添加式更新和一次写多次读的工作流特征为目标，以较高的异常发生率和较好的容错性能为基础设计的分布式文件系统。Bigtable 是 Google 公司的一个分布式结构化数据存储系统，是一个稀疏的、分布式的和持久化存储的多维度排序 key-value 存储系统。当前，在技术路线上可以分为纯 key-value 存储模型上的基于 MapReduce 技术的分析型数据管理技术、MapReduce 与数据库技术相结合的分析型管理技术和以数据库为中心融合 MapReduce 功能的混合型管理技术。从未来的发展趋势看，MapReduce 在海量数据非结构化处理方面具有优势，而数据库在结构化高性能数据处理方面具有优势，将二者在功能层面上进行融合能够更好地结合两类技术各自的优势，为用户提供统一的分析型查询处理平台支持。

参考文献

1. Abadi D J, Madden S R, Hachem N. Column-stores vs. row-stores: how different are they really? SIGMOD Conference, 2008: 967-980

2. Jeffrey Dean, Sanjay Ghemawat. MapReduce: simplified data processing on large clusters. OSDI, 2004: 137-150

3. Stonebraker M. SQL databases vs. NoSQL databases. Commun. ACM, 2010, 53(4): 10-11

（王珊　陈红　张延松）

fenxing

分形（fractal）　没有特征尺度、具有某种自相似性的图形和结构的总称。所谓特征尺度是指某一事物在空间或时间上反映本质属性的特定度量。例如，在几何学中，通常的长度、面积、体积就是几何特征尺度。在流体力学中，水力直径就是一种特征尺度。在集成电路中，特征线宽就是一种特征尺度。分形一词由数学家 B. B. Mandelbrot 在 1975 年创造，它来自拉丁文 frāctus，有“不规则”“破碎”之意。将分形看作具有如下性质的集合 F：

（1）具有精细结构，即在任意小的比例尺度内包含整体。

（2）不规则的，不能用传统的几何语言来描述。

（3）通常具有某种自相似性，至少是近似的或者统计意义下的自相似性。

（4）在某种方式下定义的“分形维数”通常大于 F 的拓扑维数。

（5）常常由非常简单的方法定义，可能由递归、迭代产生。

自然界里类似分形的事物有云、山脉、闪电、海岸线、雪花、植物根、树冠、西兰花、动物的毛皮的图案、血管、大脑皮层、岩石的断裂口等等。

根据自相似性的程度，分形可分为如下三类：

严格自相似：这类分形在任一尺度下都呈现完全一样的形态。例如，Cantor 三分集、Koch 曲线等。

近似自相似：这类分形在不同尺度下会显得大致相同，包含整体缩小尺寸的某种扭曲及退化形式。

统计自相似：这类分形在不同尺度下都能保有固定的数值（如分形维数）或统计测度。例如，蜿蜒曲折的海岸线、漂浮的云、布朗运动等。

分形几何学（fractal geometry）是一门以分形这类非规则几何形态为研究对象的几何学，被称为描述大自然的几何学。研究分形性质及其应用的分形理论（fractal theory）现已成为非线性科学（nonlinear science）的一个重要分支；而分形、混沌（chaos）、孤立子（soliton）是非线性科学中三个最重要的概念。分形理论的本质是一种新的世界观和方法论。自然界中，很多物体形态是无序、不稳定、不规则和随机的，这些都是传统几何学难以描述和表达的。分形理论使人们能以新的观念、方法和手段对此进行描述和表达，并透过无序的混乱现象和不规则的形态，揭示隐藏在复杂现象背后的规律，以及局部和整体之间的本质联系。

如今，分形几乎无处不在。分形理论的应用遍及数学、物理、化学、材料科学、生物与医学、地质与地理学、天文学、计算机科学、管理，乃至经济、社会、艺术等各个领域。

在分形几何学中，几何对象的空间维数不一定是整数，可以是分数或者是连续变化的实数。分形维数（fractal dimension）是分形几何学最重要的参量，它能定量地描述分形集合的复杂性和不规则度。下面通过 Koch 曲线的产生过程来简要说明分形维数。

设 E_0 是单位长直线段；E_1 是由 E_0 除去中间 1/3 的线段，而代之以底边在被除去的线段上的等边三角形的另外两条边所得到的图形，它包含四个线段；对 E_1 的每个线段都进行同一过程来构造 E_2，以此类推。于是得到一个曲线序列 $\{E_k\}$；其中 E_k 是把 E_{k-1} 的每一个直线段中间 1/3 用等边三角形的另外两边取代而得到的。如图 1 所示。

当 $k\to\infty$ 时，曲线序列 $\{E_k\}$ 趋于一个极限曲线 F，并称 F 为 Koch 曲线。

Koch 曲线 F 是分形曲线，它是自相似的，四个

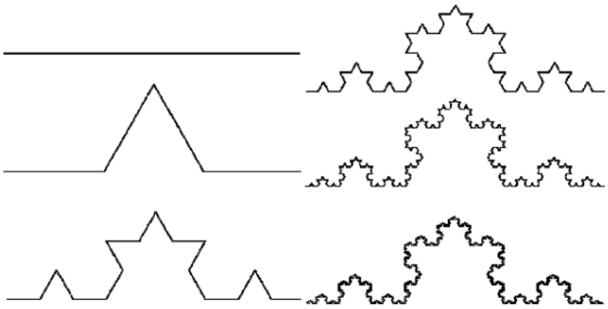


图 1　Koch 曲线的形成过程

部分与整体的相似比例为 1/4；F 具有精细结构，即在任意小的比例尺度内包含整体；F 是不规则的，不能用传统的几何语言来描述；F 的长度为 $l(F)=\lim\limits_{k\to\infty} l(E_k)=\lim\limits_{k\to\infty}(4/3)^k=\infty$，而面积为 0。

因为 F 的长度为∞，而面积为 0，所以 F 的维数既不是 1，也不是 2，而是一个介于 1 与 2 之间的分数。下面用相似性维数（一种分形维数）求 F 的维数。

设某图形是由将原图缩小为 $1/a$ 的相似的 b 个图形所组成，如果 $a^D=b$，即 $D=\ln b/\ln a$ 成立，那么 D 称为该图形的相似性维数。

D 可以是整数，也可以是分数。对 Koch 曲线 F 来说，首次是把 F 分成 4 个部分，每个部分都是原来大小的 1/3，而每一部分又可以同样地继续再细分；因此 $a=3$，$b=4$，于是 Koch 曲线 F 的相似性维数或分形维数为

$$D=\frac{\ln b}{\ln a}=\frac{\ln 4}{\ln 3}\approx 1.2619$$

分形维数有很多种定义，除了相似性维数，还有 Hausdorff 维数、Bouligand 维数、容量维数、盒维数（或计盒维数）、关联维数、信息维数等。

分形理论的发展大致可分为三个阶段：

第一阶段为 1875 年至 1925 年，在此阶段人们已认识到几类典型的分形集。例如，Weierstrass 函数曲线，此曲线特点是“处处连续，但处处不可微”；能填满一个平面区域的填充曲线：Peano 曲线和 Hilbert 曲线；Cantor 三分集、Koch 曲线、Sierpinski 三角、Sierpinski 地毯、Sierpinski 海绵、Brown 运动等。并且，人们力图对这类集合与经典几何的差别进行描述、分类和刻画。

第二阶段大致为 1926 年到 1974 年，人们对分形集的性质做了更深入的研究，特别是维数理论的研究获得了丰富的成果。这些研究深化了第一阶段的思想，不仅逐渐形成理论，而且将研究范围扩大到数学的许多分支中。1967 年，Mandelbrot 在美国

"Science"杂志上发表题目为"How long is the coast of Britain? Statistical selfsimilarity and fractional dimension"(《英国的海岸线有多长？统计、自相似和分数维数》)的论文,标志着其分形思想萌芽的出现。

第三阶段为1975年至今,是分形几何在各个领域的应用取得全面发展,并形成独立学科的阶段。1975年,Mandelbrot在巴黎出版的法文著作"Les objets fractals: forme, hasard et dimension"(《分形:形状、机遇和维数》),这部著作将前人的工作进行总结,集其大成,并正式提出分形的概念,第一次系统地阐述了分形几何的思想、内容、意义和方法,标志着分形几何作为一门独立的学科正式诞生。1977年,Mandelbrot在美国出版其英文版"Fractals: Form, Chance, and Dimension"(《分形:形状、机遇和维数》)。1982年,Mandelbrot对英文版"Fractals: Form, Chance, and Dimension"进行增补,更名为"The Fractal Geometry of Nature"(《大自然的分形几何》)出版。这部著作引起了各国学者的广泛关注,并迅速形成了"分形热",分形几何在自然科学、社会、经济和艺术等各个领域的应用蓬勃发展起来。

分形理论的发展离不开计算机图形学的支持,如果一个分形构造的表达,不用计算机的帮助是很难让人理解的。

分形理论的发展正方兴未艾,很多理论问题还有待进一步研究,但分形理论至少在以下三方面给人们以启示:首先,自然界中许多不规则形态的背后都有规则,分形整体与局部形态的相似启发人们通过认识部分来认识整体,从有限中认识无限。其次,许多以前被认为是随机的现象,从分形理论的角度看并不是随机的,比如布朗运动、股票价格的波动、传染病的流行传播等,这为人们控制这些貌似随机的现象奠定了理论基础。最后,分形揭示了介于整体与部分、有序与无序、复杂与简单之间的新形态、新秩序;分形理论中的分形维数概念,为人们认识世界中的复杂形态提供了一个新的尺度和对复杂性做定量分析的工具。

参考文献

1. Mandelbrot B B. The fractal geometry of nature. New York: W. H. Freeman Company, 1982

2. [英]肯尼思·法尔科内. 分形几何——数学基础及其应用. 5版. 曾文曲,刘世耀,戴连贵,等译. 沈阳: 东北大学出版社, 2003

3. 谢和平,薛秀谦. 分形应用中的数学基础与方法. 北京: 科学出版社,1977

4. 朱华,姬翠翠. 分形理论及其应用. 北京: 科学出版社,2011

(朱晓临)

fenzhi xianjie fa

分支限界法(branch-and-bound approach)

一种一般的用于求最佳解的算法设计方法。分支限界法是一种结合"限界"技术的最小代价优先结点生成法。即按照最小代价优先的方式从活结点表中选取扩展结点进行扩展,并且及时使用"限界"技术将不可能产生所需解的活结点变成死结点。

最小代价优先的基本思想是:为了克服深度优先和宽度优先搜索状态空间树的盲目性,定义$C(X)$表示在以X为根的子树上有最小代价的回答结点的代价。期望每次从活结点表中选取$C(X)$的值最小的活结点作为扩展结点(即优先展开代价最小者)。但是,在构造出以X为根的完整子树之前无法计算出代价函数$C(X)$的精确值。因此,只好采用估值函数$L(X)$来代替$C(X)$。也就是说每次从活结点表中选取$L(X)$的值最小的活结点作为扩展结点。可以证明:只要采用的估值函数$L(X)$对所有结点X均满足$L(X) \leqslant C(X)$并且结束时活结点表中结点的估值$L(X)$均大于或等于已找到的最佳解的代价,那么最小代价优先法找到的一定是真正的最佳解。上述终止条件意味着找到一个解后,不能立刻停机,而应继续到所有活结点的估值均大于或等于已找到的解的代价。估值函数$L(X)$的选取决定了算法的性能。虽然$L(X)$用于代替$C(X)$,但是并非$L(X)$越接近$C(X)$越好。即使对所有结点X都有$L_1(X) \leqslant L_2(X) \leqslant C(X)$,采用估值函数$L_2(X)$的算法也未必比采用估值函数$L_1(X)$的算法要快。因为重要的不是$L(X)$的绝对大小而是其相对大小关系是否准确地反映了$C(X)$的相对大小关系。事实上,若对任意结点$X$和$Y$满足:$L(X) \leqslant L(Y)$当且仅当$C(X) \leqslant C(Y)$,则性能最佳。

"限界"技术的基本思想是:为了缩短活结点表的长度,定义上界函数$U(X) \geqslant C(X)$。令$U = \min\limits_{X已生成}\{U(X)\}$,若活结点表中的结点$X$满足$L(X) > U$,则从活结点表中删除$X$。基于$L(X) > U$从活结点表中删除结点$X$,不会导致生成的结点数增加。因为假设$X_i$是$X$的儿子结点,那么$U(X_i) \geqslant C(X_i) \geqslant C(X) \geqslant L(X) > U$。这样即使让$X$扩展出$X_i$,也不会因此而使$U$的值变小。一般来说,上界函数$U(X)$的值在不小于$C(X)$的前提下越小越好。因为这时

U 也小,从而删除的活结点就多,执行速度更快。

给定下界函数 $L(X)$ 和上界函数 $U(X)$ 使得 $L(X) \leqslant C(X) \leqslant U(X)$。对解结点 X,$C(X)$ 可计算。下面是使用分支限界法求最佳解的一个一般的算法框架。其中,LEAST(L)返回活结点表 L 中 $L(X)$ 的值最小的结点 X。PARENT(X)记录状态空间树中结点 X 的父结点。

```
if Root 是解结点 then begin U:= C(Root); ans:=
  Root; end else begin U:= U(Root); ans:=0; end
L:= {⟨Root,L(Root),0⟩};
while L≠∅ do
  begin
    E:= LEAST(L); 设结点 E 的满足约束函数 B
      的全部儿子结点为 x1,x2,…,xn;
    for i:=1 to n do
      if xi 是解结点且 C(xi) ≤ U then begin U:=
        C(xi); ans:= xi; end else if U(xi) < U then
        U:= U(xi);
    L:= (L - {⟨x,L(x),PARENT(x) > |L(x)⟩U})
      ∪{⟨xi,L(xi),E⟩|1≤i≤n 且 L(xi)≤U};
  end;
print ('least cost = ', U); while ans ≠ 0 do begin
print (ans); ans:= PARENT(ans); end;
```

作为一种一般的算法设计方法,分支限界法可用于求解各种各样的问题(如旅行商问题、带时限的作业选择问题、15 迷问题等)。

参考文献

1. Aho A V, Hopcroft J E, Ullman J D. The design and analysis of computer algorithms. Boston, MA: Addison-Wesley, 1974

2. Cormen T H, Leiserson C E, Rivest R L, et al. Introduction to algorithms. 3rd ed. Cambridge, MA: The MIT Press, 2009

3. Kleinberg J, Tardos E. Algorithm design. Boston, MA: Addison-Wesley, 2005

(殷建平)

fenzhi fa

分治法(divide-and-conquer approach) 一种一般的算法设计方法。分治法的基本思想是:把一个大问题,分成若干较小的子问题,求解这些子问题,最后使用这些子问题的解求出原问题的解。

由分治法产生的子问题往往与原问题类型相同,但规模变小,这样便可反复应用分治法,直到子问题小到可直接求解。因此,使用分治法常常获得一个递归算法。将大问题分成小问题的分法在很大程度上决定了设计算法的性能。一般说来,为了获得高效的算法,分解时通常遵循平衡原则,即尽量使子问题规模相等,特别是采用二等分。若希望获得最坏情况下的复杂性比较好的算法,则使子问题的规模尽量相同。当问题规模不是整幂时,可采用“规模”整幂化或近似等分法。前者将原问题转化为某个等价的规模是整幂的问题,后者允许子问题的规模有所不同,但力求使其差异尽可能小。若希望获得期望复杂性比较好的算法,则使子问题的规模在概率的意义下尽量相同。使用“分治法”时,算法所需时间是将原问题分解成子问题所需时间、解各子问题所需时间与根据子问题的解来求解原问题的解所需时间之和。

作为一种一般的算法设计方法,分治法可用于求解各种各样的问题(如最大子数组问题、计数逆序问题、最近点对问题、卷积与快速傅里叶变换问题)。特别是,不但可用于求解以比较判定为主的问题(如排序问题、顺序统计问题),获得最坏情况下复杂性(如合并排序算法)或期望复杂性(如快速排序算法)比较好的算法甚至最佳算法,而且可用于求解以计算为主的问题(如整数乘问题、矩阵乘问题),甚至可用于求解某些“难解问题”(如马的周游路线问题)的部分解。

参考文献

1. Aho A V, Hopcroft J E, Ullman J D. The design and analysis of computer algorithms. Boston, MA: Addison-Wesley, 1974

2. Cormen T H, Leiserson C E, Rivest R L, et al. Introduction to algorithms. 3rd ed. Cambridge, MA: The MIT Press, 2009

3. Kleinberg J, Tardos E. Algorithm design. Boston, MA: Addison-Wesley, 2005

(殷建平)

fenzu chuanshuwang

分组传输网(packet transport network, PTN) 一种以分组为传送单位,以承载电信级以太网业务为主,同时兼顾 TDM、ATM 和帧中继等业务的综合传送技术,又称分组传送网。是针对解决基础传输网络,尤其是城域网汇聚接入层如何同时高效支持 TDM 和分组数据等综合业务的问题而提出的。PTN 在 IP 业务和底层光传输媒质之间设置

了一个功能层,针对分组业务流量的突发行为和统计复用传送的要求来进行设计,以分组业务为核心并支持多业务提供,增加了适应数据业务的特性:分组交换、统计复用、采用面向连接的标签交换、分组 QoS 机制、灵活动态的控制面,同时保留了光传输网络的传统优势,包括高可用性和可靠性、高效的带宽管理机制和流量工程、良好的网络扩展性、丰富的操作维护(OAM)、快速的保护倒换等。

电信级以太网,又称运营商级以太网(carrier Ethernet, CE),是由城域以太网论坛(Metro Ethernet Forum, MEF)在 2005 年年初提出的。城域网以太网论坛是由 35 家公司发起成立的,其目的是进一步明确和强调以太网技术在运营商网络中的应用和价值,旨在把以太网变成能让电信运营商在城域网内使用的技术,提供与传统电信网络在 QoS 保证、安全、可运营可管理等方面具备相同保证能力的以太网。电信级以太网是在保留传统以太网帧结构的基础上,通过添加传送网功能、扩展帧头或引入二层协议和信令等方式,在以太网上实现和电信网类似的可管理性和可靠性。根据 ITU-T 和 MEF 的定义,电信级以太网具备以下特征:①高可靠性。在环型、双星型和格型拓扑下能够提供 50 ms 以内的保护和自愈能力;②端到端的 QoS 保障能力。具备业务区分和识别能力,能够提供基于 CIR 和 EIR 的 QoS 保障能力;③完善的 OAM 和可管理性。基于二层提供对故障和性能的管理功能,具备灵活的业务管理和提供能力;④多业务。能够满足 TDM、话音和视频等业务的综合承载需求,通过伪线或仿真方式实现和现有网络的互通;⑤标准化。具备良好的互联互通性,实现不同厂商和运营商的业务互通。

多业务传送平台(multi-service transport platform, MSTP)

多业务传送平台(MSTP)是指基于 SDH 技术同时实现 TDM、ATM、以太网等业务的接入、处理和传送,并由提供统一网管的多业务节点构成的数据传送平台。MSTP 节点除应具有标准 SDH 传送节点所具有的功能外,还具有以下主要功能特征:①具有 TDM 业务、ATM 业务或以太网业务的接入功能;②具有 TDM 业务、ATM 业务或以太网业务的传送功能,包括点到点的透明传送功能;③具有 ATM 业务或以太网业务的带宽统计复用功能;④具有 ATM 业务或以太网业务映射到 SDH 虚容器的指配功能。

参考文献

1. 李芳,张海. 分组传送网(PTN)的生命力探讨. 通信世界, 2008, (37)

2. 徐荣,龚倩,邓春胜,田沛. 电信级以太网. 北京:人民邮电出版社, 2009

3. 曹蓟光,吴英桦. 多业务传送平台(MSTP)技术与应用. 北京:人民邮电出版社, 2003 (张健)

fenzu jiaohuan

分组交换(packet switching) 存储转发交换技术中的一种。它是将要传送的报文分割成许多具有统一格式的报文分组,简称分组,并以此为传输的基本单元进行存储转发。按对分组传输路径的管理不同,分组交换又可分为虚电路和数据报两种方式。从物理上看,逻辑电路所经过的物理信道同时可为其他通信用户所共享。也就是说,物理信道可被多

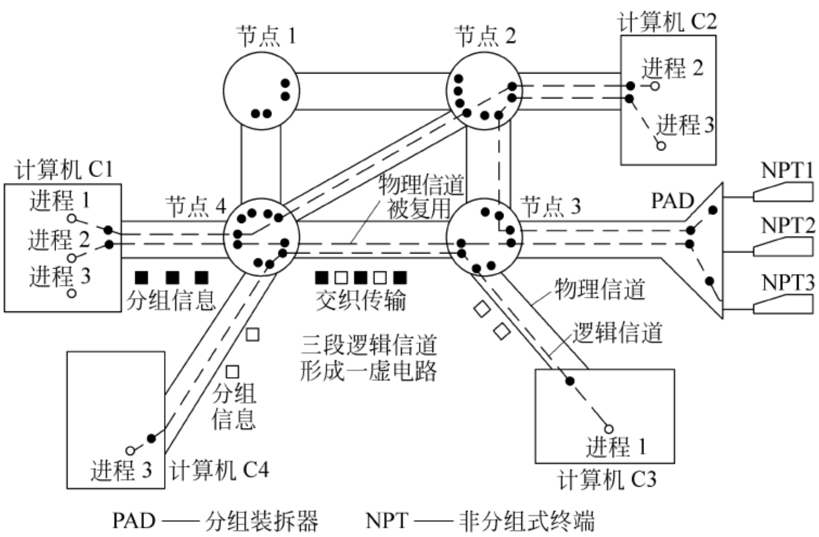


图 1 分组交换工作原理

对通信用户复用,物理信道上交织传输着不同用户对的分组信息,如图 1 所示。

*虚电路*是通信双方在进行通信时首先在它们之间建立一条逻辑电路,在一次通信中,该电路为通信双方所独占,这一点和**电路交换**相同。虚电路是在每次通信时通过通信双方呼叫和协商建立,然后传送数据,通信完毕即行拆除,这个过程称为一次虚呼叫。这种虚电路称为*呼叫虚电路*(CVC),也称为*交换虚电路*(SVC)。如果通信双方的虚电路是被永久分配定的,则这种虚电路称为*永久虚电路*(PVC)。无论是 SVC 还是 PVC,它们都是由多段逻辑信道组成。物理信道可以分为多个逻辑信道,构成多个虚电路,从而实现物理信道的复用。在使用虚电路的情况下,一次通信中的各分组都是沿着该条虚电路传输的。虚电路的例子有 X. 25、帧中继和异步转移模式 ATM 等。

数据报的每个分组传输路径不是沿着固定的一条虚电路,而是每经过一个交换机就要动态地选择一次,这同报文交换的存储转发交换方式路径选择相似,故称为*数据报*。目前,分组交换主要是指基于数据报技术,使用 TCP/IP 协议的互联网 Internet。

参考文献

1. 高星忠,陈锦章,张有材. 分组交换. 北京:人民邮电出版社, 2001

2. Behrout Forouzen, 等. 数据通信与网络. 潘乞,朱丹宇,周正康,译. 北京:机械工业出版社, 2000

（史美林　徐明伟）

fengzhuang

封装(encapsulation)　一种用以隔离书写程序中所作的某些决断的技术。这些决断有数据表示、算法实现等。为此,往往将程序组织成接口和内部两部分,只是接口对用户可见,内部对用户是隐蔽的。

封装的作用有四:一是提供一种信息隐蔽技术;二是用以强制推行特定的访问风纪,例如,通过管程访问程序的临界区;三是在程序间提供某种相容性,后者是通过一个接口来实现的(接口将控制和数据转换成一种所封装的程序可用的形式);四是对特别难用的资源提供抽象,例如,程序人员可以使用的计算机的有些控制功能是通过操作系统接口提供的,这样,既安全又便于使用。

参考文献

Ralston A, et al. Encyclopedia of computer science. 3rd ed. Wiley, 2003

（徐家福）

fengzhuang nei xitong

封装内系统(system in package, SiP)　将多个芯片、元器件及其互连装配在同一个封装体内而形成的微系统。与**片上系统**(SoC)一样,SiP 也是一种微系统的集成方式。但 SoC 通常只能集成同一种工艺类型的多个知识产权核(IP 核),而 SiP 可以集成不同工艺代(如 0. 13 μm 和 90 nm)、不同工艺类型(如互补金属氧化物半导体(CMOS)和锗硅)的多个芯片,不仅更加灵活,而且设计验证的复杂度低、产品上市时间短。

在同一个封装管壳内集成多个芯片的工作始于 20 世纪 60 年代。一开始是把晶体管等有源元件与电阻、电容等无源元件安装在基板上,再进行封装,称为混合集成电路。后来又把多个集成电路芯片及一些无源元件安装在高密度多层互连基板上,再统一封装,称为多芯片模块(multi-chip module, MCM)。MCM 按其高密度互连基板的制造工艺可分为 3 种:①MCM-L,基板是多层层压印制电路板(PCB);②MCM-C,基板是陶瓷或玻璃瓷,通过高温共烧陶瓷法(HTCC)或低温共烧陶瓷法(LTCC)制成;③MCM-D,通过硅或介质材料上的淀积布线形成基板互连。在计算机领域中较早采用 MCM 来提高组装密度、电性能和可靠性的是 IBM 公司的大型机系统。为了解决组装密度提高后带来的散热问题,IBM 3081 等大型机中还引入了热导模块。IBM 公司近年推出的 POWER4/5 高性能微处理器也采用了 MCM 技术。

与把多个封装好的芯片直接安装在 PCB 上相比,MCM 在提高组装密度上是有明显优势的。但是,MCM 里,多个芯片是以二维方式排列在基板上的,如 IBM POWER5 的 MCM 里,平面排列了 4 个 POWER5 芯片和 4 个容量为 36 MB 的三级 cache 芯片。这种二维分布芯片的形式,限制了封装尺寸的进一步缩小。为了满足平板电脑、智能手机等设备对元件尺寸的苛刻要求,出现了芯片堆叠式的 MCM,即 SiP。因为存储器芯片在尺寸、接口等方面的一致性,最早的 SiP 就是把多个存储器芯片堆叠在一起再封装起来,形成一个容量更大但占地面积较小(也许稍厚一点)的部件。后来,又出现了把 CPU、DRAM、Flash 存储器和其他器件堆叠在一起的更高效更复杂的 SiP。为了区别于典型的复杂 SiP,有时把堆叠少量同种类(如 DRAM 和 Flash 存储器)芯片的多芯片封装称为 MCP(multi-chip package)。通常,MCP 内各芯片间基本上没有信号传递和数据

交换，更像一个多芯片构成的部件。而典型SiP内各芯片间存在较复杂的控制和通信关系，更像一个系统或子系统。

SiP中芯片之间的互连，主要有两种方式。第一种方式是通过引线键合连接到封装上。引线键合在集成电路的封装中经常使用，用在SiP中碰到的难点主要是引线密度的增加以及芯片堆叠时存在的高度差。另一种方式主要适用于倒装片（flip chip），即利用倒装片上的焊球把相邻堆叠的芯片直接连接起来。这种方式不仅能提供大量的互连，而且可通过缩短连线来提高性能，但对工艺的精密度也要求更高了。

通过SiP可以把构成一个电子系统的不同类型集成电路（如CMOS处理器、DRAM、Flash存储器、光电子传感器、锗硅射频电路等）组合到一个封装体内。由于对参与SiP集成的各个芯片可先进行功能验证和测试，因此SiP器件可达到较高的成品率。基于这样的SiP器件，再添加少量零部件，就可以形成一个完整的电子系统，可应用于各类手持移动终端。由芯片堆叠而形成的SiP在散热方面并不具有优势，而终端设备用的芯片普遍具有低功耗的特点，正好适合在SiP集成。技术和需求的契合，促进了SiP的发展。

SiP技术还在发展之中。主要的发展方向包括：集成更多芯片、改进芯片间的互连、增强散热能力、完善设计与测试方法、拓宽应用领域等。

参考文献

金玉丰，王志平，陈兢. 微系统封装技术概论. 北京：科学出版社，2006

（唐志敏）

fuwu gongcheng

服务工程（service engineering） 按照满足用户要求和创造最大服务价值的原则，运用服务科学和**服务计算**的理论和技术对服务与服务系统进行描述与定义、设计与优化、实现与实施、运行与维护/重构的工程。服务工程是一种多学科的工程技术和方法体系，是系统工程方法在服务领域中的应用与扩展，从工程学的角度来设计最佳的服务及服务系统实现。它在一组工程化的模型、方法体系、原则、工具和平台环境的支持下，从服务系统生命周期、服务控制机制和顾客需求等三个角度研究服务系统的优化设计与构建，支持更好地设计与实现服务系统，管理好服务系统生命周期，使服务系统具备优化性能，提高顾客满意度。

服务工程形成于20世纪末和21世纪初，其技术源泉主要来自三个方面：面向服务的软件工程、服务科学与工程、现代服务业的应用发展。①软件系统服务化带来了复杂异构、协同交互、按需应变、网络服务等新特征，使传统软件工程方法逐步发展为面向服务的软件工程，进而形成了服务工程。1996年被提出的**面向服务的体系结构**（SOA）是面向服务的软件工程的典型代表。②2004—2005年，IBM公司提出了“服务科学、管理与工程（service science, management and engineering, SSME）”概念，从服务科学的视角提出了服务工程，促进了服务工程理论技术体系的形成与发展。③21世纪以来，现代服务产业的快速发展催生了越来越多的复杂服务系统，迫切需求一种服务工程对服务业务创新和服务系统设计实现提供理论与技术支持。

服务工程的核心内容包括：

（1）服务系统　定义各类服务要素（人、软件、硬件、环境、技术、共享信息等）及其关系的体系结构、系统配置、运作机理。

（2）服务生命周期　指服务与服务系统存在的各个阶段，如服务需求获取与定义、服务模型描述与建模、服务系统设计开发、服务系统实现与构建、服务系统运行维护等。

（3）服务方法论　面向服务系统生命周期，对服务系统进行描述、定义、设计、建立、实施、运行、维护、重构、评价。

（4）服务优化控制　评估服务系统实际运行情况与顾客期望之间的差距，通过反馈、沟通、自我调整、修改优化等手段对服务系统进行控制与调整。

（5）服务顾客评价　获取与描述服务系统顾客的需求/期望，评价客户对服务系统运行的满意度，度量服务系统向顾客提供的价值。

服务工程的技术内容主要包括：服务方法论、服务模式创新与管理、服务生命周期管理、服务体系结构、服务模型驱动体系结构、服务表示与建模分析、仿真与优化、服务需求工程、服务语义学、服务价值理论、服务质量优化设计、服务解决方案与服务提交、服务系统开发与实现、服务优化与重构、服务构件及服务组合、服务评价（如功能、性能、价值、质量、信用、风险等评价）与管理、服务工程支撑环境及辅助工具等。由于服务系统是为服务供需双方创造最大化的服务价值，在服务工程中，人们注重研究服务价值理论与价值模型、面向价值的服务方法论及建模方法、价值知觉的（value-aware）服务方法论、

价值约束下的服务组合与实现技术等。

在服务工程中,服务方法论居重要核心位置。服务工程方法论是指一套用于对服务系统进行架构与规划、描述/建模与设计、构建与实施,并对服务系统性能进行优化改进的方法与规则,为服务系统设计人员提供方法指南,使之有章可循。服务方法论的主要技术内容包括:

(1) 服务模型　以形式化、结构化或图形化形式表示服务创新设计思想、业务需求、服务系统形态,刻画各类服务要素(包括资源、能力、人员、行为、过程等)及其关联关系;同时,还可表示服务要素的语义信息或约束。

(2) 服务建模方法　使用恰当的服务建模语言和规则,按照特定的设计方法和步骤,完成服务系统的需求分析、设计和实施模型的转换与映射。通常不同的服务模型规范具有特定的建模方法。为了支持服务建模,还涉及服务模型语义增强、模型转换与互操作、模型分析与验证等。

(3) 服务系统构建方法　根据服务模型,利用服务系统开发平台或工具构建相应的服务系统。面向服务的体系结构(SOA)和基于服务构件的服务组合技术是经常被用于服务系统的构建。

(4) 服务性能评价　对即将或者已经建立的模型、服务系统及其性能(如顾客满意度、功能、性能、质量、价值、成本、信用、风险等)进行评价,支持对服务系统的判断和优化调整。

(5) 支撑工具与平台　支持服务工程的 IT 工具及平台,包括服务建模工具、服务系统开发工具、服务系统部署支持工具、服务系统运行架构及平台、服务评价工具等。

(6) 服务系统实施指南　将已开发的服务系统部署到应用实际运行环境中的方法与手段,包括对相关软件、硬件、数据、网络、人员等相关要素的配置说明、具体实施步骤以及方法指南等。

典型的服务方法论分为四类:①在传统**软件工程**方法上发展起来的服务方法论,如面向服务的建模与体系结构(SOMA)、面向服务的应用系统开发方法(SODA)、面向服务的统一过程(SOUP)等;②模型驱动的服务方法论,与软件领域的模型驱动体系结构(MDA)类似;③基于领域工程的方法论,如基于框架的服务方法论、可配置的服务方法论等;④语义驱动的方法论。

随着现代服务业的迅猛发展和传统产业服务化的转型,服务工程及方法论在社会服务、生产服务等不同领域得到广泛应用,例如现代服务业、产品制造相关服务、信息服务、软件服务、IT 服务管理、**云计算**服务、智慧城市服务、智慧交通服务、务联网与**物联网**服务、**电子政务**服务等。虽然它们面向的服务领域和服务对象均不同,但都遵循相似的服务工程哲理和方法论。服务工程为各类服务系统的优化设计与实现发挥着重要作用。

当前,服务工程的新发展主要围绕服务新模式、服务系统体系结构、服务价值、服务质量、服务信用等服务新特征来展开。未来的主要技术发展趋势为:①面向服务模式创新的服务工程　将服务新模式作为服务工程的源头,通过模型驱动方法支持服务创新的实现;②面向价值的服务工程　将服务价值作为服务系统的主要指标,提出服务价值理论与价值模型,发展价值知觉的服务方法论;③面向复杂服务系统协同的服务工程　面对大规模定制顾客服务需求支持复杂服务协同业务和系统实现;④面向顾客满意度的服务质量工程　基于顾客期望,运用服务质量展开方法进行服务优化设计与实现;⑤基于信用的服务工程　进行服务信用评价与控制,支持服务信用体系的优化设计与实现;⑥面向情境的服务工程　建立面向应用情境的服务模型及服务语义学,发展情境感知的服务工程;⑦云计算环境下的服务工程　充分运用云计算平台的基础设施即服务(IaaS)、平台即服务(PaaS)、软件即服务(SaaS)、商务过程即服务(BaaS)等服务机制实现务联网(IoS)的万物皆服务(EaaS)等。

参考文献

1. IEEE Computer Society's Technical Committee on Service Computing (TCSC). http://tab.computer.org/tcsc/

2. 徐晓飞,王忠杰. 服务工程与方法论. 北京:清华大学出版社,2011 年

3. European Commission ICT Research in FP7. http://cordis.europa.eu/fp7/ict/home_en.html

(徐晓飞　王忠杰)

fuwu jisuan

服务计算(services computing)　利用信息服务与计算技术来有效地表示、创建、运作与管理商业服务的新技术。它跨越了计算机与信息技术、商业管理、商业咨询服务等领域。服务计算覆盖了服务及其创新的全生命周期,消除了商业服务与信息支撑技术之间的鸿沟,其目标是在信息技术与计算技

术支持下使商业服务更加有效和高效。

目前,关于“服务”和“服务计算”的概念有两种理解。一是将“服务”看作各领域业务应用,把“服务计算”理解为一种宏观/中观层面的商务管理哲理与技术,以“面向服务”的视角看待社会、经济、组织等商务服务系统,用计算的方法和技术解决服务领域的各类问题,分析、设计、实现、运行并优化服务系统,提供用户满意的服务;二是将“服务”看作开放自治和与平台无关的网络构件,把“服务计算”理解为微观层面的新型分布式计算,以“面向服务”的方法构造分布式异构软件服务及其集成系统,用面向服务的思维和方法解决软件需求、设计、开发、演化过程中的各类问题,并通过基于服务基础设施与运行环境的服务系统来实现计算系统与商务业务的协同。

追溯服务计算的产生发展起源,主要有三条线索:①大规模分布式异构复杂软件系统的发展,从软件建模方法、**软件体系结构**、软件实现等技术角度催生了**Web 服务**、**面向服务的体系结构**(SOA)和服务计算。②现代服务产业的蓬勃发展和软件即服务(SaaS)的出现,从社会应用角度催生了万物皆服务(EaaS)和服务计算。③现代商务应用的动态性和复杂性要求按需服务和业务流程持续优化,从商务运作角度催生了面向服务的体系结构、**云计算**和服务计算。其中,1996 年被首次提出的面向服务的体系结构(SOA)可以被看作是直接导致服务计算概念产生的起源。SOA 强调软件系统的去耦合、基于开放标准互操作、大粒度重用、低成本和快速开发、动态扩展等特征,可使分布式软件应用具有更好的复用性、灵活性和可增长性。这些都是服务计算所具有的良好特性。Web 服务是早期阶段最有代表性的 SOA 实现技术。2003 年 11 月,IEEE 服务计算技术执行委员会成立,正式将“服务”与“计算”结合在一起,标志着“服务计算”这新兴技术领域开始独立发展。此后,服务计算逐渐从单纯的网络化软件技术向“业务 + 软件”相结合的方向发展,**物联网**、社会信息网络和云计算的兴起都大大促进了服务计算的发展。

根据 IEEE 服务计算技术执行委员会发布的服务计算技术内容以及人们近年来关于服务计算的研究方向,可以将服务技术的内容归纳为以下几个方面:

(1) 服务与服务系统的表示与建模技术　从宏观商务服务层面和微观计算层面研究服务与服务系统及其生命周期的模型与表示,揭示服务与服务系统的机理与发展规律;主要包括:服务的概念及模型、服务的性质及特征、服务与服务系统生命周期、服务需求工程、**服务工程**与服务方法论、服务价值理论及价值知觉的服务方法论、服务建模技术与服务模型驱动方法论、服务语义学等。

(2) 面向服务的计算实现技术　从微观计算层面采用面向服务的体系结构和计算方法支持分布式异构软件服务及其集成系统的实现与运行,为计算系统与商务业务服务的协同提供信息技术支持;主要包括:面向服务的体系结构 SOA、Web 服务、网格计算与效用计算、服务发现与服务组合技术、服务构件实现技术(如 Web service、SCA、SDO 等)、服务关系实现技术(如 BPEL 等)、支持服务实现与运行的协议(SOAP、UDDI、WSDL、WS-系列协议等)、软件即服务(SaaS)、服务基础运行设施(如企业服务总线(ESB)等)、**云计算**技术、服务协调运作方法与动态演化理论、Web 2.0/3.0 技术、务联网(IoS)技术等。

(3) 服务创新与业务流程集成技术　从宏观商务服务层面描述服务业务创新、流程分析优化、咨询与交付的方法,并给出利用信息技术支持商务服务从服务需求分析到服务交付过程的集成方法与技术;主要包括:服务模式创新设计与管理、面向服务的咨询分析与优化方法论、服务业务过程管理与集成、服务交付平台与方法、面向服务的业务模型与建模方法、服务业务模型到 IT 模型的转换与优化、基于服务构件与服务组合的业务实现、基于云计算架构的万物皆服务(EaaS)、服务质量管理与改进方法等。

(4) 服务应用解决方案及管理技术　从服务应用角度给出领域服务解决方案及服务管理的相关技术及标准化;主要包括:领域应用服务与标准、服务系统管理技术(含服务监管、安全、隐私、信任、事务、协调、优化、保护等)、服务系统的可信性管理(含服务信用与风险管理等)、信息技术服务管理、服务性能管理与评价、基于移动计算的服务解决方案、典型现代服务业的行业领域服务解决方案及参考模型等。

由于服务计算通过提供先进的商业服务模式和面向服务的 IT 使能技术,可以帮助企业或组织机构在商务业务层与 IT 技术层灵活高效地进行交互协同,提供令顾客满意的高效率、高质量服务,使得服务计算有着十分广泛的应用领域。在现代服务业、各种行业商务服务应用系统、**互联网**服务应用系统、

云计算与物联网(IoT)/务联网(IoS)、大型分布式计算系统体系架构、电信及信息服务业、电子商务(参见**电子商务系统**)、电子政务(参见**电子政务系统**)、**智能交通**、智能电力、智慧城市等行业和领域发挥着重要作用,也为许多传统制造业向服务型制造企业的转型升级产生着重要影响。

经过近十年的发展与积累,服务计算的基础理论方法和技术体系已经初步构建完成,并对计算技术的服务化趋势产生着重要影响。目前,服务计算技术也在不断向广度与深度发展,其主要研究趋势有:①服务个性化　面对日益复杂的大规模个性化顾客需求,服务计算系统必须提供复杂服务集成与协同方式支持个性化定制服务,以保证最大顾客满意度。②服务动态化　针对顾客需求和应用的不断变化,服务系统通过模型驱动等方式进行自适应式动态演化或快速动态地进行资源与服务重构,保持服务系统最佳运行性能和有效服务协同。③普适泛在化　随着移动计算、云计算、物联网、务联网的不断发展,未来互联网环境下的服务系统将使人们能够随时随地通过各类移动设备以客户个性化、高效快速的方式进行服务交互与交付,便捷高效地使用服务。④内容智能化　未来的服务系统将服务资源、数据资源和知识内容等实现智能分类、分布与管理,通过**数据挖掘**、信息检索、资源发现等智能化手段,根据客户潜在需求提供主动服务内容的推荐与定制。⑤资源虚拟化　未来互联网环境下的服务系统把现实世界物理资源和数字世界信息资源进行最大限度的有效整合,并通过虚拟化技术实现资源的发现、集成、协同、管理与利用,以云服务方式支持跨时空的服务交付与提供。⑥应用领域化　随着服务计算应用的不断普及,各类服务资源、内容和业务将根据应用领域越分越细,形成越来越多的垂直领域细粒度服务,并面向领域实现各种创新的服务模式,满足各应用领域的客户更细微的需求。

参考文献

1. IEEE Computer Society's Technical Committee on Service Computing(TCSC). http://tab.computer.org/tcsc/

2. 张良杰,张嘉,蔡弘. 服务计算. 北京:清华大学出版社,2007

3. 徐晓飞,王忠杰. 论服务计算与服务工程的发展及影响. 中国计算机科学技术发展报告(2008). 北京:清华大学出版社,2009

4. 徐晓飞,王忠杰. 服务工程与方法论. 北京:清华大学出版社,2011

(徐晓飞　王忠杰)

fuwuqi

服务器(server)　指在网络环境中或在具有客户-服务器结构(参见**客户-服务器计算**)的分布式处理环境中,为客户的请求提供服务的节点**计算机**,或指在该计算机上运行的,用于管理资源并为用户提供服务的**计算机软件**。服务器的两种定义可能会引起混淆,例如"Web服务器"可以指用于提供Web服务的计算机,也可以指提供这些Web服务的软件程序。在该名词出现时具体使用哪种定义需要根据上下文予以确定。

客户-服务器是实现资源共享的一种结构,客户是服务器的服务对象。在某种应用环境中的服务器,也可能会在另外一种应用环境中成为客户。在不同的应用环境下,这里的"客户"与"服务器"可以用于指代通过通信网络连接的两台自治的计算机(或其上运行的互相协作的软件程序),也可以用于指代同一台计算机上运行的具有不同功能的互相协作的程序。

作为服务器的计算机可以是**微型计算机**、**工作站**、**小型计算机**、**大型计算机**乃至**大规模并行处理**的高性能计算机。服务器软件主要包含**操作系统**、**网络协议**、**数据库管理系统**以及各种开发工具与**软件中间件**。这些软件用来支持客户和服务器之间相互作用,负责透明地连接客户和服务器系统。服务器可提供文件、数据库、打印、通信、图形、图像、安全、保密、系统管理、网络管理以及信息发布等服务;按服务器的规模和性能可分为群组级服务器、部门级服务器和企业级服务器等,按服务器用途可分为专用服务器和通用服务器,此外尚有按客户与服务器之间的连接是否通过通信服务器而分为远程服务器和近程服务器。

服务器的主要特点有:①服务器只是在客户的请求下才为其提供服务,而不主动为客户提供服务;②透明性　即服务器对客户完全透明,一个与服务器通信的客户完全不必知道服务器的存在及其工作情况;③高性能、高速度、大容量、高可靠性及可伸缩性。

服务器的概念最早出现于**局域网**条件下的客户端-服务器结构。这时的服务器按功能可分为文件服务器、打印服务器、数据库服务器、应用服务器和通信服务器等,应用服务器又可分为计算服务器、决策支持服务器、联机事务处理服务器等。

文件服务器是服务器中最早出现的一类，它主要为用户提供文件存储和共享服务，文件服务器一般具有较强的存储能力和较高的通信速度，借此为客户提供高速和大容量的文件存储服务。**打印服务器**则是用于打印服务的专用服务器，通过使用打印服务器的服务，客户端可以共享高速打印资源。数据库服务器是服务器中专门提供数据库服务的一类服务器，它为网上客户提供数据库的查询、更新、索引、事务管理等服务。通信服务器则是为客户端提供各种通信转发、中继等方面服务的服务器。

随着互联网的发展和普及，作为互联网一个重要组成部分的互联网服务器也日益受到重视。通过互联网向分布在全球的用户提供各类信息服务的服务器叫作**互联网服务器**。互联网服务器可以提供的服务包括将按树状结构分层的域名解析为互联网地址的域名的地址转换服务（DNS）、完成电子邮件的传输与管理的电子邮件服务（E-mail）、文件传输服务（FTP）、电子公告板服务（BBS）、广域信息服务（WAIS）、网络新闻服务（USENET）、万维网服务（Web，WWW）等。一台互联网服务器可以同时提供一种或多种上述服务。（黄震春）

fuwu qianyi

服务迁移（service migration）　指挂起现有执行实体，停止源端机器上的服务程序，将服务的运行时数据、执行实体的状态迁移到一个目的机器，并在目的机器上恢复服务的运行。服务迁移可以实现集群或数据中心内机器的维护和升级、负载均衡、容错、绿色计算等。

服务迁移的方式分为两种，即离线迁移和在线迁移。

离线迁移，也称为静态迁移，只是简单地将关闭或挂起的服务进行了迁移。由于只和静态的内存数据、磁盘数据相关，因此离线迁移较容易实现。

在线迁移，也称为动态迁移，是指迁移一个运行中的服务，包括不断变化的内存信息和磁盘数据。在线迁移的目的是保持服务的连续不间断性，即在服务迁移完成之后，目的端上的服务能在源端挂起的检查点状态上继续运行。服务的在线迁移能快速透明地将一个运行中的服务从一个机器传输到另一个机器，整个迁移过程对服务所需的操作系统、相应的应用程序以及远程客户端透明，唯一能察觉到的变化是，服务在迁移过程中会有轻微的性能下降并在迁移完成之后出现性能的提升，因为服务迁移到目的端后通常会得到更多的系统资源。在线迁移能满足许多企业的需要，是服务迁移的主要方式。一般而言，服务迁移指的是服务的在线迁移。

服务迁移主要涉及以下几种资源的迁移：内存的迁移、文件系统的迁移以及外围设备的重定向。

内存迁移是服务迁移的重要部分。内存信息涉及服务的所有执行状态，具有动态性。内存迁移是确保服务迁移状态一致性的关键，并影响整个服务迁移的性能。目前，内存迁移大多采用预复制的方法。在源端机器仍然提供服务的同时，首先将全部内存页面数据传输到目的端，然后通过一轮轮的循环反复将内存中新修改的页面从源端机器复制到目标机器，期望被修改的页面越来越少，当新修改的页面数少于一定的阈值，或者新修改的页面数不再减少，或者循环次数达到一定的阈值时，挂起源端机器上的服务，将最后一轮修改的页面和相关机器状态传输到目的端，在目的机器上恢复服务的运行。预复制方法的目的是通过一轮轮的页面复制来尽可能减少服务的宕机时间（迁移过程中服务完全不可用的时间），但它不可避免地造成一部分页面的多次传输，既增加了服务迁移的延迟，又浪费了网络带宽资源，造成服务质量的下降。为避免预复制方法存在的内存页面冗余传输现象，有研究者提出后复制的方法来实现内存迁移，首先将服务的机器状态传输到目的机器，并在目的机器上启动服务的运行，接着传输内存页面到目的机器，这样，可以保证所有内存页面在服务迁移过程中只被传输一次。另外，针对传输的页面数据量大的问题，也有研究人员提出了基于检查点、跟踪/回放技术的内存迁移方法，不断跟踪、记录源端服务的非确定性事件并传输到目的机器，并在目的机器上进行服务执行状态的回放，和预复制方法不同的是，每一轮传输的不再是内存页面数据，而是服务的执行状态日志，从而大幅度减少了迁移过程中传输的数据量。

文件系统的迁移传输可以采用写时复制技术，使用哈希函数法判断重复的数据块，仅发送不相同的数据块。由于存储系统通常涉及的数据量巨大，数据中心一般采用网络存储系统，从而避免文件系统的迁移。

外围设备的重定向主要保证网络连接不受影响，在**局域网**，可以使用地址解析协议（ARP）将 IP 地址（参见 **Internet 地址**）直接定向到新的物理机器上；而在广域网，可以采用 IP 隧道和动态域名系统（DNS）结合的方法实现网络的重定向。

服务迁移具有如下特点：①服务迁移是动态迁移。服务迁移过程中,服务没有中断,保证服务的连续性。②服务迁移具有透明性。迁移对服务相关的系统软件和应用程序是透明的,不需要重新设置或者编程等,网络的重定向也是通过保持连接状态下进行 IP 地址的改变。③服务迁移的前后状态具有一致性。迁移结束后,服务在目的机器上开始执行时,它的状态和在源机器上停止时的状态是一致的。④服务迁移是有一定系统代价的。迁移过程中,服务质量会有所下降。服务迁移会额外增加对资源的需求,如网络带宽、处理器周期、本地硬盘的吞吐率等。

参考文献

1. 金海,等. 计算系统虚拟化——原理与应用. 北京：清华大学出版社,2008

2. Milojicic D, Douglis F, Paindaveine Y, et al. Process migration. ACM Computing Surveys, 2000. 32(3)：241-299

3. Hines M R, Gopalan K. Post-copy based live virtual machine migration using adaptive pre-paging and dynamic self-ballooning. Proceedings of the 5th ACM/USENIX Conference on Virtual Execution Environments (VEE' 09), ACM Press, 2009：51-60　(吴松)

fuwu rongqi

服务容器(service container)　提供服务构件运行环境的一种软件。服务构件通常指 Web 服务构件,服务容器一般指 Web 服务容器。

Web 服务容器的体系结构一般可以分为传输层、消息层、服务层和适配层四个层次。传输层主要处理各种传输协议,如 HTTP 协议、HTTPS 协议、SMTP 协议等;消息层主要处理各种消息协议,如 SOAP 协议;服务层主要处理 WSDL 协议,确定请求的报文所匹配的 Web 服务及其服务方法;适配层主要调用具体的 Web 服务实体,通过统一接口的服务适配器,提供一致的 Web 服务调用机制。

服务构件一般以服务部署包的形式部署到服务容器中,服务容器负责将新服务添加到服务注册表中,并对其进行初始化。当服务调用请求到来时,服务容器的适配层通过查找服务注册表获得所请求的服务实例,并调用该服务实例获得返回结果。为了对服务容器及容器中服务构件进行监控管理,服务容器一般都包含一个监控管理模块,该模块一方面不断搜集服务容器的实时状态信息发送给容器内的一个称为管理客户端的控制程序,另一方面还可以直接执行管理客户端发送的管理命令,从而实现对服务容器的监控和管理。

服务容器的概念来源于 1996 年出现的**面向服务的体系结构**(SOA)。SOA 中基于服务的概念定义了一种系统集成方法,使得软件构件能够以服务的形式发布、发现、绑定和调用。随着互联网应用的普及,为了能够将应用业务包装成独立的、可伸缩性的基于互联网的服务,业界提出了 Web 服务的概念。国际标准化组织 W3C 认为：Web 服务是一个通过 URI 识别的软件应用程序,其接口及绑定能用 XML 文档来定义、描述和发现,并且使用基于 Internet 协议上的消息传递方式与其他应用程序进行直接交互。目前 Web 服务的接口普遍采用 Web 服务描述语言 WSDL 来描述,其他应用程序一般通过 SOAP 消息机制与 Web 服务实现交互。Web 服务容器通过统一的技术规范,支持基于互联网的 Web 服务构件的部署、运行、调度和管理,成为 SOA 运行环境的最佳实践。　(刘旭东)

fuwu zhiliang baozheng

服务质量保证(quality assurance of service)

通过适当的管理手段,为用户所请求服务的质量提供某种程度的保证。服务质量指服务对外表现出的非功能属性,它决定了用户对服务的使用体验。

服务质量的概念最早于 1994 年由国际电信联盟给出其在通信领域定义,主要包括网络连接中的各方面需求,比如服务响应时间、差错率、回声,中断等需求。在计算机网络和分组交换领域,服务质量是指为不同的用户、应用或数据流提供不同的优先级,或者对数据流保证某种程度的性能,比如保证所需的比特率、延迟、抖动、丢包概率和(或)误码率。在分布式软件系统中,服务质量不仅与服务本身有关,也与其所处的网络环境有关,可以分解为：①服务器端所提供服务本身的质量,如平均响应时间、安全性、可用性、可靠性以及与具体业务相关的其他指标等;②网络传输过程中的服务质量,可以量化为吞吐量、差错率、端到端延迟、延迟抖动等指标。

分布式软件系统中的服务质量保证研究可以分为两个方面。一方面,对服务提供者而言,主要是如何对其所提供服务的质量进行有效管理,以满足用户的需求。服务等级协议是较为常见的手段,它是服务提供者和客户之间就服务质量、优先级和权责达成的一致性契约,在此基础上可以通过逆向计费

或其他机制来支持服务质量保证。另一方面,从服务使用者而言,主要挑战是如何从已有的海量服务中进行选择和组合,以达成特定应用所需的服务质量,相应研究涉及服务质量信息的收集和查询、质量敏感的服务组合方法等。

近年来,许多关键领域大型分布式软件系统的部署,对软件服务提出了较高的质量保证需求。云计算的发展促使服务从传统的硬件转移到云端。这些服务需要和传统的服务提供同样甚至更高要求的服务等级,目前,服务等级协议在云服务中应用前景广泛。云计算运行环境的开放性和动态性也为服务质量保证技术的研究带来了新挑战。

参考文献

1. 韩蕾,蔡皖东,何得勇. 面向 Web 服务器集群系统的 Web QoS 机制研究. 计算机科学,2004,31(8):32-34

2. Papazoglou M P. Web 服务:原理和技术. 龚玲,张于涛,译. 北京:机械工业出版社,2010

(刘旭东)

fuwu zuhe

服务组合(service composition) 将多个软件服务组织在一起,按特定顺序执行,以实现某一业务目标或者提供一个新的服务的过程。软件服务是对软件资源进行封装,对外提供一定功能并且可以被动态发现和调用的自治实体。服务组合对于服务使用者而言,只需与组合后的服务交互,从而简化了使用,且便于快速构建大规模的分布式软件系统;对于服务开发者而言,实现了服务的有效重用,能够以较低代价实现可演化。

服务组合的研究内容包括服务的描述、发现与选择、服务的组合过程与组合结果的验证以及服务的在线演化等。其中,服务组合过程中业务流程的创建通常可以分为编制(orchestration)与编排(choreography)两个方面。①编制指一个可执行的业务流程,可与流程内部和流程外部的服务进行交互,从一方角度刻画了控制。交互发生在消息层,包括业务逻辑与任务执行顺序。基于编制的方法,存在一个集中的协调者,负责调用各个服务,完成服务组合。②编排记录了各方之间的消息序列,描述了 Web 服务之间的消息交互。基于编排的方法,不存在集中的协调者,各个参与组合的服务是对等的,通过对服务之间交互协议的严格定义来实现服务组合。

服务组合目前应用最为广泛的是 Web 服务的组合。Web 服务目前所采用的技术标准包括简单对象访问协议 SOAP,Web 服务描述语言 WSDL 以及通用描述、发现与集成服务 UDDI 等。SOAP 为基本服务的互操作性定义了 XML 消息传递协议,WSDL 采用了用于描述服务的通用语法,UDDI 为系统地发布和发现服务提供了所需的基础结构。这些规范一起共同使应用程序能够遵循一个松散耦合、与平台无关的模型来找到对方并进行交互。

在开放、动态多变的互联网环境下,软件服务普遍以 Web 服务的形式包装。服务组合能够实现灵活的资源聚合和应用集成,是互联网环境下的分布式软件系统构造的主要方式之一。在基于编制的组合语言方面,IBM、微软等提出的服务组合语言 BPEL4WS 现已成为 OASIS 组织标准。BPEL4WS 定义了用来描述基于流程及其相关方之间互操作的业务流程行为的模型和语法,并定义了这些相关方之间的多重服务互操作是怎样被协调起来达成业务目标的以及这种协调所需的状态和逻辑。它能够描述由 Web 服务参与的复杂业务流程,同时又能将多个服务包装成为更高级别的 Web 服务并发布出去,目前已经在电子商务等领域得到了应用。在基于编排的组合语言方面,W3C 组织提出了一种基于 XML 的 Web 服务编排描述语言 WS-CDL,它从全局的角度定义了参与者共同的行为和补充可见的行为,即有序的消息交换最终导致完成共同的业务目标,进而描述了参与者的端到端的协作,因此能够支持服务组合过程多个服务的对等协同。 (刘旭东)

fudianshu biaozhun

浮点数标准(standard for floating-point numbers) 由标准化组织制定的关于浮点数表示格式和运算规则的标准。

浮点数是实数的一种近似表示,浮点数格式必须兼顾表示范围及表示精度的要求,浮点数运算规则必须尽可能地保持精度、减小误差。在计算机诞生后的很长一段时间里,由于没有统一的浮点数标准,不同品牌、不同系列的计算机采用各不相同的浮点表示形式和运算规则,给数值计算和软件移植带来了困难。考虑到**微处理器**性能的不断提高和计算机应用的进一步普及,IEEE(电气和电子工程师协会)在 20 世纪 80 年代制定了浮点数标准,成为所有微处理器遵循的二进制浮点算术运算标准,即 IEEE 754-1985 标准。

IEEE 754 标准于 1985 年 3 月获 IEEE 标准委员会批准，同年 7 月成为美国国家标准。1989 年成为国际电工委员会标准（IEC559：1989）。IEEE754 标准的内容包括浮点数的表示形式、浮点操作的类型和定义、舍入方式、例外处理方法等。这里主要介绍浮点数的表示形式。

IEEE 754 主要定义了单精度（32 位）和双精度（64 位）两种基本格式以及扩充单精度和扩充双精度两种扩充格式，但对扩充精度仅指定了对精度的最低要求。计算机系统可以用硬件、软件或硬软件结合的方式实现 IEEE 754 标准或标准的主要部分，且任何实现都必须至少包括单精度浮点格式。

在 IEEE 754 的浮点格式中，尾数用原码表示，指数用移码表示，各种格式的有关参数见表 1。

表 1　IEEE 754 标准定义的浮点格式参数

参数	单精度	扩充单精度	双精度	扩充双精度
表示精度（位）	24	≥32	53	≥64
最大指数	+127	≥ +1 023	+1 023	≥ +16 383
最小指数	−126	≤ −1 022	−1 022	≤ −16 382
指数偏移	+127	未指定	+1 023	未指定
指数部分位数	8	≥11	11	≥15
格式总位数	32	≥43	64	≥79

基本格式由 1 位符号 s、指数部分 e 和小数部分 f 组成，如图 1 所示。对单精度数，e、f 分别是 8 位和 23 位；对双精度数，e 和 f 分别是 11 位和 52 位。左边是最高位。此格式表示的数 X 的值 v 由如下规则确定。

s	e	f

图 1　IEEE 754 标准定义的浮点格式

对单精度数：

（1）若 $e=255$ 且 $f\neq0$，则无论 s 是什么，v 称为 NaN，即非数（Not a Number）；

（2）若 $e=255$ 且 $f=0$，则 $v=(-1)^s\times\infty$，即正无穷或负无穷；

（3）若 $0<e<255$，则 $v=(-1)^s\times2^{e-127}\times(1.f)$，称为规格化数；

（4）若 $e=0$ 且 $f\neq0$，则 $v=(-1)^s\times2^{e-126}\times(0.f)$，称为反规格化数；

（5）若 $e=0$ 且 $f=0$，则 $v=(-1)^s\times0$，即零。

对双精度数：将上面的 255，127，126 分别替换成 2047，1023 和 1022 即可。

上述定义表明，除了零有正零（+0）和负零（−0）两种形式外，其他表示格式对应的数值都是唯一的。用移码表示指数的一个好处是：比较非负浮点数的大小时，可以把它们当成整数来处理。

大部分浮点数都采取由（3）定义的规格化数的形式，其尾数的最高位为一隐含的 1，从而 23 位小数部分可表示 24 位精度。作为一种特例，（4）既解决了（3）不能表示零的问题，又能充分利用有限的格式空间表示更多非常接近于零的数，以减小计算误差。由（4）表示的数称为反规格化（denormalized）数。（1）和（2）中定义的∞ 和 NaN 是特殊值，其中∞ 可能来自被零除或上溢，NaN 可能是 $\sqrt{-1}$ 这类实数运算中非法操作的结果。当结果为特殊值时，标准允许由具体实现来决定是继续运算还是中止运算并产生例外。在允许继续运算的实现中，任何由 NaN 参与的运算，结果都是 NaN，而由∞ 参与的运算，仿照对函数求极限的方法进行（如 $1/\infty=0$，$\arctan(\infty)=\pi/2$ 等）。当极限不存在时，运算结果为 NaN。由于在（1）中定义的 NaN 只要求 f 为非零值，在允许产生例外的实现中，f 可用于存放系统状态或例外原因，供例外处理程序使用。

IEEE 754 标准规定的舍入方式共 4 种，即向最近数、向 +∞、向 −∞ 以及向 0 舍入，其中向最近数舍入是默认的舍入方式。在默认情况下，若运算结果与最近的两个可表示数的距离相等，则取最低位为零者（即舍入至最近的偶数）。

除 754 标准外，IEEE 在 1987 年还颁布了 854 标准。虽然被称为“进制无关的浮点算术运算标准”，但 IEEE 854—1987 只涉及二进制和十进制这两种形式，且没有规定具体的数据格式，因而影响不大。2008 年，IEEE 公布了集 754—1985 和 854—1987 于一体并有所扩展的 754—2008 标准，除兼容原有的单精度、双精度这两种基本格式外，新标准还定义了 16 位的半精度和 128 位的四精度二进制浮点数格式及 32/64/128 位的十进制浮点数格式，另外，新标准将反规格化数改称为亚规格化（subnormal）数，含义更为准确。

参考文献

1. IEEE. 754—2008：IEEE Standard for Floating-Point Arithmetic. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp? arnumber=4610935

2. Goldberg D. What every computer scientist should know about floating-point arithmetic. ACM Com-

puting Surveys, 1991, 23(1): 5-48　　　　(唐志敏)

fudian yunsuanqi

浮点运算器(floating-point unit)　计算机内专用于执行浮点运算的功能单元。浮点数是实数的一种近似表示方法,典型的浮点运算包括加、减、乘、除、倒数、开方等(参见**浮点数标准**),有些系统的浮点运算器还支持某些超越函数(如三角函数与幂函数等)。配备浮点运算器的计算机又被称为浮点计算机,以区别于只支持定点运算的定点计算机。高速浮点运算能力是现代通用计算机的最重要特征之一。

定点计算机中的浮点运算都是用由定点指令组成的子程序来模拟的,效率十分低下。所以一些20世纪50年代中后期问世的大型计算机(如IBM704)就开始装备专用的浮点运算硬件。到60年代,以IBM360/91的流水线浮点运算器为标志,浮点运算器的设计技术基本成熟,并逐步进入中小型系统。

由于集成度不够的原因,早期的**微处理器**或者不包含浮点运算器(如Intel 8008),或者由单独的芯片提供浮点运算功能(如Intel 8086微处理器与8087浮点协处理器)。集成度提高后,通用微处理器都内置了浮点运算器,但一些对低成本低功耗有明确要求或应用场合只需要定点运算的处理器(如部分微控制器和数字信号处理器),仍仅支持定点运算。

从浮点数的表示形式可知,浮点运算既涉及指数(阶码)部分的运算,又涉及小数(尾数)部分的运算。这些单独的运算实际上都是定点运算,因此概念上浮点运算器就是由阶码部件和尾数部件这两种定点部件组合而成的。尾数部件实质上就是一个通用的定点运算器,能实现加、减、乘、除、移位、舍入等基本算术运算。对阶码部件的要求更简单一些,只要能进行阶码相加、相减和比较操作即可。图1给出了浮点加、减运算的大致流程。

因为浮点运算的步骤比较多、延迟比较长,为了提高性能,常采用流水线的工作方式来提高其吞吐率。例如,浮点加减法可按对阶(阶码比较)、右移(阶码小的尾数右移)、尾加(尾数相加)、舍入、左规(通过左移形成满足标准格式要求的浮点结果)这五大步骤分成五个流水级。当然,流水级的划分方案是多种多样的,比如,根据所用的浮点运算算法、工艺技术条件和工作频率规划,也可以把浮点加减法划分为对阶右移、尾数相加、舍入左规三个流水级,等等。

有时为了节省器件,可将浮点加减和乘法的功能组合在同一个流水线中,形成所谓的多功能流水线。由于加减和乘法会使用多功能流水线中流水级的不同子集,需要专门的控制逻辑,避免加减操作与乘法操作竞争某个流水级。在追求高性能的系统(如多发射处理器)中,常为浮点加减与浮点乘法分设独立的流水线,以尽可能提高运算的并行度。

浮点运算中可能出现的异常类型比定点运算多,其处理复杂度也远高于定点运算。复杂的异常处理不必完全由浮点运算器的硬件实现,而通常是由操作系统中的异常处理程序配合完成的。

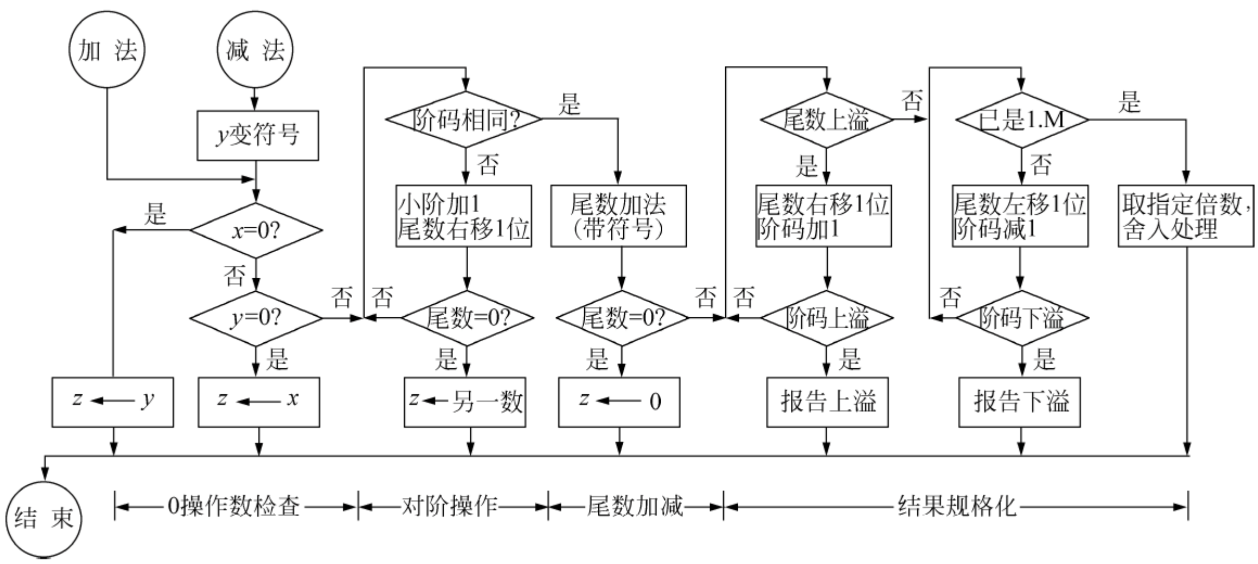


图1　浮点加、减运算流程

除用于**中央处理器**外，浮点运算的工作原理还应用在向量处理器、图形处理器、高性能数字信号处理器等场合。

参考文献

1. Patterson D A, Hennessy J L. Computer organization and design: the hardware/software interface. 北京:机械工业出版社,2010

2. 徐洁,俸远祯. 计算机组成原理与汇编语言程序设计. 2 版. 北京：电子工业出版社,2005

（张悠慧　唐志敏）

fuhao xuexi

符号学习(symbolic learning)　研究如何从大量观测到的符号数据中归纳出一般判别规则和模式的理论、模型与方法。

符号学习可以追溯到 20 世纪 50 年代末 Bruner 等人关于人类概念学习过程的研究、Samuel 的下棋程序以及 Chomsky 和 Solomonoff 关于文法的研究。基于结构主义语言学，Gold 以系统、严格的方式研究了文法学习的问题，其结果表明，在文法学习的问题上存在着不可逾越的障碍。随后，在 20 世纪 60 年代，人们将所处理的数据的表示限制为“属性-值”形式，因此，相对于基于自然语言数据的学习，以这种数据为基础的学习得到了简化。其中，Hunt 的 CLS(Concept Learning Systems)与 Michalski 的 AQ 是颇具代表性的两类学习算法。由此，自 70 年代始，符号学习逐步形成了两个重要的系列：ID 与 AQ。

AQ 系列的学习建立在**命题逻辑**的运算之上，它使用“永假与命题的析取等于命题，永真与命题的合取等于命题”，以及吸收率来简化给定的数据集，最后的蕴涵式就是最终学习到的规则集。

ID 系列的学习建立在数据集的划分之上，其中 Quinlan 在 1986 年给出的 ID3 算法是这类方法成为**机器学习**关注热点的关键。ID3 与以后发展的 C4.5 目前仍然是常用的方法。ID3 的重要贡献在于，这个算法将信息熵作为树结构生长的准则，其本质是从属性集中寻找能够更好地分类样本的属性。

Reduct 理论源自波兰数学家 Pawlak 1992 年提出的 Rough sets，其本质与符号学习一致。AQ 系列算法可以理解为这个理论的一个特例；同样，ID 系列算法也可以基于这个理论描述。不过，其对信息熵的刻画并不自然。

总体上，符号学习的优点在于归纳的规则和学习到的模式具有可解释性，一般用于数据描述、数据分析以及基于用户需求的个性化求解。

参考文献

1. 陆汝钤. 人工智能(下). 北京：科学出版社,1996

2. 洪家荣. 归纳学习——算法　理论　应用. 北京：科学出版社,1997

（韩素青）

fudu tiaozhi jishu

幅度调制技术(amplitude modulation technology)　幅度调制技术根据信息源信号的类型分为数字调幅(amplitude-shift keying, ASK)和模拟调幅(amplitude modulation, AM)。

模拟调幅是使高频载波信号的振幅随调制信号的幅度变化而变化的调制技术。在调制过程中，高频载波的幅度随时变化，而载波信号的频率保持不变。

数字调幅，又称幅移键控法，是用高频载波信号的两个不同振幅来分别表示调制信号的两个二进制数值“1”和“0”。在有些情况下，用一个振幅恒定的载波的存在表示一个二进制数字；而用一个振幅为零(或载波不存在)表示另一个二进制数字。

先设定载波信号如下式

$$A_c(t) = A\cos(2\pi f_c t + \theta)$$

其中振幅为 A，载波频率为 f_c，相位为 θ。

使用信息源信号 $a(t)$ 对载波信号　进行幅度调制的时候，调制信号用以下公式表示

$$s_I(t) = a(t)\cos 2\pi f_c t$$

式中的下标 I 表示对同相载波已调制过，并且此时认为振幅 A 等于 1。若式中 $a(t)$ 为连续函数时，便为模拟调幅 AM；若式中 $a(t)$ 取值离散时，便为数字调幅 ASK。图 1 为 ASK 的详细表示图。

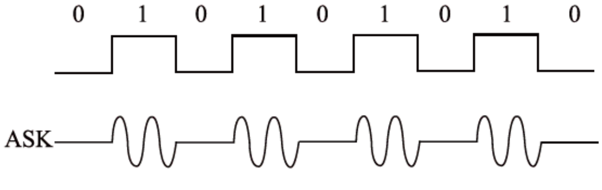


图 1　幅移键控法 ASK

参考文献

1. [日]关清三. 数字调制解调基础. 北京：科学出版社, 2002

2. 蒋青，于秀兰. 通信原理. 北京：人民邮电出版社, 2008

（董守斌　张凌　张效祥）

fushedu jishu

辐射度技术(radiosity technique) 基于热能辐射原理,在封闭环境中求取物体表面光能分布的一种全局光照技术(参见**光照模型**)。该技术于1984年提出。它特别适用于由漫反射面组成的封闭环境。辐射度技术是根据光能能量传播平衡的原理计算环境中每一平面片上的光照能量。利用该技术可以很自然地形成物体之间的颜色渗透现象。

辐射度技术所考虑的是具有各向相同的反射特性的漫反射物体,即表面不甚光滑的物体。这些物体对于光照在各个反射方向上具有等量均匀的反射光亮度,因而辐射度计算的辐射光亮度与观察者的位置(即视点)无关。经典的辐射度技术是将环境中的平面(曲面)划分成足够小的面片,而后根据环境中光能传播平衡的原理建立线性方程组求解每个面片上的光强。

光能传播和辐射计算过程中所涉及的一个重要问题是,对于一个特定的具有光能的面片,它所传播出的光能中究竟有多少比值的能量能够到达另一特定面片。在辐射度技术中这一比值称为形状因子。形状因子的大小只与环境中物体的几何状况(大小、位置)有关,因而一旦造型环境确定,形状因子即可预先计算。通常在辐射度求解其中形状因子的计算占用绝大部分开销,因而形状因子的计算效率对于辐射度计算举足轻重。在1985年提出了一种效率较高的形状因子计算技术,称为半立方体方法。在利用该方法求取从一个面片(如面片i)到其他面片的形状因子时,在此面片中心放置一个单位半立方体,然后将其他面片向半立方体各面上进行投影,以求取形状因子的大小。

作为一种成功有效的全局光照技术,辐射度技术自1984年以来在多方面取得重大进展。首先,为了使辐射度技术解决复杂环境中的应用问题,1988年出现了逐步求精的辐射度技术;为了使辐射度技术能够处理包括非漫反射物体的环境,提出了多种非漫反射环境的辐射度技术;在辐射度求解的过程中,环境物体需划分成作为求解单元的小面片,为了提高效率和改善图形质量,提出了有效地组织和划分面片的层次辐射度技术。常规的辐射度技术是针对平面(曲面)环境设计的,而将该技术推广到更复杂的环境尤其重要。当前辐射度技术已经成功地推广到具有参与介质(云雾、灰尘等)、几何纹理面(橘子皮等)、分数维几何面(山脉等)、毛绒表面等复杂环境。

参考文献

1. 唐荣锡,汪嘉业,彭群生,汪国昭,等. 计算机图形学教程. 修订版. 北京:科学出版社,2000

2. Cohen M F, Wallace J R, Hanrahan P, Greenberg D P. Radiosity and Realistic Image Synthesis. San Francisco, CA: Morgan Kaufmann, 1993 (吴恩华)

fuzhu cunchuqi

辅助存储器(secondary memory) 用于长期存放大量暂时不运行的程序和数据的存储设备。又称为外部存储器(简称外存),属于计算机的外围设备。CPU不能直接访问外存。常见的辅助存储器有磁盘、磁带、光盘、U盘等。在辅助存储器中,**磁盘存储器**的存取速度快、存储容量大,是主要辅助存储器设备。传统的磁盘主要是磁介质存储,目前还有一种采用固态电子存储芯片阵列作为存储介质的**固态盘**(solid state disk, SSD),固态盘具有启动快、读取延迟小等特点。**磁带存储器**的存取速度慢,但易于脱机存放,通常作为离线存储器用来存放需要长期保存的信息。**光存储器**的存储容量大,存取速度接近磁盘存储器,但它不易写入,多采用只读的工作方式,用于存放大量程序和数据。**U盘**是采用**快闪存储器**(flash memory,简称Flash或闪存)进行存储的,其优点是便于携带、读写方便,目前已经成为移动存储的主要介质。

数据在辅助存储器中是以记录块为单位进行存取的。在存取时,按中央处理器给出的块地址找到所需的记录块,用成块方式与主存储器交换数据。按照寻找记录块的方法,辅助存储器的存储设备分为两类:①顺序存取存储器。存储器的读写机构从所在的记录块开始,顺序查找,直到找到所需要的块,磁带存储器属于这一类。②直接存取存储器。存储器的读写机构按记录块的地址直接寻找到一个较小的区域,然后在这个区域内顺序寻找所需的记录块。例如,在磁盘和光盘存储器中先找到柱面,再在柱面内找到所要的块。直接存取存储器的存取速度比顺序存取存储器的快。

为了进一步提高硬盘的容量、速度及可靠性,独立冗余磁盘阵列(redundant array of independent disk, RAID)应运而生。RAID的基本思想是把多块独立的硬盘组成一个硬盘组(逻辑硬盘),并通过冗余来提高可靠性。RAID技术主要包含RAID 0到RAID 7等规范,它们的主要区别在于数据在**磁盘阵列**上的分布方式及采用的容错技术。

参考文献

Stallings W. 计算机组织与体系结构. 7 版. 张昆藏,等译. 北京:清华大学出版社,2006

(郑衍衡　陈妍　王换招)

fuwang cunchu

附网存储(network attached storage,NAS)　一种将存储设备直接连网并使用标准协议提供文件级数据访问的存储方式。在传统方式中,数据请求经过服务器发给存储设备,存储设备通过服务器上网。采用附网存储方式的存储设备自身具有网络接口,能直接连接到诸如**以太网**等各种网络。标准协议是指网络文件系统(NFS)、通用互连文件系统(CIFS)等。文件级数据访问指数据请求是以文件句柄及偏移值为参数的访问方式。附网存储的突出特点就是在物理连接上将存储器直接连接到网络上,不再挂在服务器后端,避免了给服务器增加输入输出(I/O)负载。

1992 年,美国加州大学的 Randy H. Katz 教授首次提出了附网存储。在以后的几年中,附网存储概念逐渐被人们所接受。由于附网存储的易扩展性与易管理性,逐步形成了一个颇具规模的附网存储产品与服务市场。早期的附网存储设备比较简单,通常把通用服务器去掉显示系统,并和存储介质集中到一个机箱中,再加上远程控制系统。后来,采用嵌入式技术设计的专用控制板和存储设备自身容量和速度的提高,附网存储设备的性能得到了快速的提高。随着网络存储技术的发展,附网存储逐渐和其他存储技术相融合,出现了集群 NAS、NAS 网关等重要的应用。

附网存储将存储设备与服务器分离,采用瘦服务器形式,精简传统服务器各部分配置,只实现文件共享服务,因此成本相对较低。它以数据为中心,集中管理数据,从而有效利用带宽,提高网络整体性能。

附网存储的设备和系统由硬件和软件两部分构成。硬件包括附网存储控制器和存储媒体,前端提供网络接口,后端提供标准的 I/O 总线接口。软件包括附网存储控制软件,接口设备驱动程序,网络应用层协议以及相应的管理工具等。

相比单台附网存储设备,集群 NAS 是横向扩展的附网存储系统,具有容量和性能线性扩展的优势。集群 NAS 协同多个 NAS 机头(NAS head)提供高性能、高可用的附网存储服务。从整体架构来看,集群 NAS 由存储子系统、NAS 机头、客户端和网络组成。存储子系统可以采用**存储区域网络**(SAN)、**直连存储**(DAS)或对象存储设备等。

附网存储有如下特点:

(1) 可扩展性强　附网存储设备提供 RJ-45 接口和单独的 IP 地址,可以将其直接挂接在主干网的交换机或其他局域网的集线器上,通过简单的设置(例如设置 IP 地址等)就可以在网络上使用,且即插即用,在线扩容无须停顿网络。因此,与传统的服务器和直接存储设备相比,附网存储设备具有容量扩展能力强的优势。

(2) 使用和管理简单　一台附网存储设备占用一个 **IP 地址**,实质上相当于一台高性能的文件服务器,却可以大幅节省设备费用。另外,附网存储设备能完全融合已建立起来的网络设备和协议,它作为独立的数据存储设备与其他的各种服务器搭配,既保护了用户的原有投资,又将整个网络的性能提高到一个新的层次。

(3) 跨平台文件共享　以 **UNIX 操作系统**家族的 Solaris、HP-UX,以及 LINUX、FREE BSD 等作为操作系统的附网存储设备通过支持网络文件系统协议实现对附网存储设备的文件共享。**Windows 操作系统**平台下的网络文件共享则采用通用互连文件系统,从而能够实现不同网络环境下用户跨平台共享数据。

附网存储已被广泛应用于互联网服务提供商(ISP)、大中小型企业以及航空、医疗等需要高速大容量存储设备解决方案的场合。

参考文献

1. Randy H Katz. Network-attached storage systems. Proceedings of the Scalable High Performance Computing Conference-SHPCC-92, 1992: 68-75

2. 张江陵,冯丹. 海量信息存储. 北京:科学出版社,2003

(周可　王冲)

fuzaxing duliang

复杂性度量(complexity measure)　算法复杂性的定量描述。算法的复杂性可由不同的标准来衡量,例如描述算法所用语言的长度,称为描述复杂性。解决一个问题的各种算法程序的长度的下界称为该问题的**复杂性**(又称 Kolmogorov 复杂性)。但最重要的复杂性度量是执行算法所耗用的资源量。一般说来,处理规模较大的输入比规模较小的输入要耗用更多的资源,这里“资源”一词主要意指时间

和存储空间。用算法耗用资源依赖输入规模的函数来表示算法的复杂程度：

$f(x,C)$ = 输入为 x 时，算法 A 所耗费的资源量

通常称 $f(x,C)$ 为算法 C 的对输入 x 的复杂性量度。记输入规模为 x 的长度 $|x|=n$，称

$$W(n,C) = \max\{f(x,n) \mid |x| = n\}$$

为输入规模为 n 时，算法 C 的**最坏情况复杂性**。又若已知输入规模为 n 的各个输入 x 的概率分布 $p_n(x)$，则称

$$A(n,C) = \sum_{|x|=n} p_n(x) * f(x,n)$$

为输入规模为 n 时，算法 C 的**平均情况复杂性**。

对于一个问题，它有各种各样的算法，我们称 $f(n)=\min\{W(n,C)$ 或 $|$ 在一定模型下解决同一问题的各个算法 $C\}$ 为该问题的固有计算难度，或称该问题的计算复杂性。实际上，计算机理论的重要分支算法设计和分析是对各个具体问题寻找复杂度尽可能低的算法 C，并求出 $W(n,C)$（或 $A(n,C)$），它是该问题的复杂度 $f(n)$ 的上界，从而得知该问题应属于的复杂性类（参见**多项式谱系**）。同时人们研究 $f(n)$ 的下界，从而得知该问题不可能属于的复杂性类，不再去设计复杂度比下界更低的算法。

从算法分析的实践中可以看出，人们感兴趣的是 $f(n)$，$W(n,C)$，$A(n,C)$ 在 n 趋于 ∞ 时的渐近性态，它们的常数项或系数是一个比较次要的因素，人们更为关心的是它们的增长率，简称阶。下列符号用于表示函数的阶（定义中的函数都是从正整数映射到正实数）。

（1）集 $O(f)$ 中的任一函数 g，存在一个常数 $r>0$，使得对于所有的 n，有 $g(n)<rf(n)$。

（2）集 $\Omega(f)$ 中的任一函数 g，存在两个常数 $n_0, r>0$，使得对于所有充分大的 $n>n_0$，有 $g(n)>rf(n)$。

（3）集 $O(f)$ 中的任一函数 g，有

$$\lim_{n\to\infty}\frac{g(n)}{f(n)} = 0$$

（4）集 $\Theta(f)$ 中的任一函数 g，有 $g\in O(g)$ 且 $g\in\Omega(f)$。

在复杂性中最常出现的阶函数有：$\log^k n$，n^k，$n^{\log n}$，2^n 等，相应阶的函数依次称为：对数阶，多项式阶，亚指数阶，指数阶。计算机科学家发现一个问题在一种计算模型下可以用多项式阶或指数阶的算法求解，那么在别的计算模型下也可以用多项式阶或指数阶的算法求解。称为**相似性和对偶性原理**。因此算法复杂性的阶是一个独立于计算模型的，而问题的复杂性的阶又是独立于算法和计算模型的仅由问题本身所决定的重要特性。

近年来，人们鉴于最坏情况复杂性研究难有进展，转向平均情况复杂性的研究；从确定型的算法复杂性研究转向不确定的、不精确型的概率算法（例如退火算法）复杂性研究；从串行算法转向并行型算法（例如遗传算法）。这些新型算法复杂性度量定义也要作相应的调整。问题的复杂性度量有时得到一些实用上的改善，但理论上，只要计算模型没突破图灵机模型，同一问题的计算复杂性度看来无望有质的飞跃。

参考文献

1. Li M, Vitanyi P. An introduction to kolmogorov complexity and its applications. Springer Verlag, 1993

2. Balcàzar J L, Dìaz J, Gabarrò J. Structural complexity, Ⅰ, Ⅱ. Springer Verlag, 1988

3. Hong Jia-Wei. Computation: computability, similarity and duality. London: Pitman, 1986 （朱洪）

fuzaxing guiyue

复杂性归约（complexity reduction） **计算复杂性理论**中研究问题复杂性类之间关系的重要方法。复杂性归约是这一方法在计算复杂性理论中的应用和发展。S. A. Cook 于 1971 年首次使用这个方法证明了第一个 NP 完全问题。在计算复杂性理论中，用各种计算模型定义了多种复杂性归约，最常用的是图灵归约和多一归约。

图灵归约 设判定问题 Π_1 和 Π_2，S 是关于 Π_2 的假想的"算法"。关于 Π_1 的以 S 为假想"子程序"的算法 A 称作 Π_1 到 Π_2 的图灵归约。若存在 Π_1 到 Π_2 的图灵归约，则称 Π_1 可图灵归约到 Π_2，记作 $\Pi_1\leqslant_t\Pi_2$。如果进一步限制 A 具有某种计算复杂性 α，则称 A 是 Π_1 到 Π_2 的 α 图灵归约。用 $\leqslant_t^\alpha$ 表示"可 α 图灵归约"。这里在计算 A 的复杂性时不考虑假想"子程序"S 的消耗，仅把调用一次 S 算作一步。当 α 为多项式时间界限时，称作多项式时间图灵归约，记作 $\leqslant_t^p$。多项式时间图灵归约是 1971 年 S. A. Cook 提出的（参见**图灵归约**）。

复杂性归约可以用来比较问题的计算难度。取图灵归约的计算复杂性 α 足够的低。设 $\Pi_1\leqslant_t^\alpha\Pi_2$，那么如果 Π_2 有某种复杂性的算法，则 Π_1 也有这种复杂性的算法。或者反过来说，如果 Π_1 不存在某

种复杂性的算法，则 Π_2 也不存在这种复杂性的算法。在这个意义下，Π_2 不比 Π_1 容易。例如，设 A 是一个多项式时间图灵归约。假设 S 是一个多项式时间算法，那么将 S 使用的时间计算在内，A 仍是一个多项式时间算法。因此，设 $\Pi_1 \leq_t^p \Pi_2$，那么如果 Π_2 是多项式时间可解的，则 Π_1 也是多项式时间可解的；反之，如果 Π_1 不是多项式时间可解的，则 Π_2 也不是多项式时间可解的。从而，相对于多项式时间而言，Π_2 不比 Π_1 容易。设 $\mathscr{C}$ 和 $\mathscr{D}$ 是两个问题类。为了研究 $\mathscr{C}$ 是否包含在 $\mathscr{D}$ 中，只需考虑 $\mathscr{C}$ 中“最难的”问题是否属于 $\mathscr{D}$。复杂性归约为定义“最难的”问题提供了工具。如果 $\Pi \in \mathscr{C}$ 并且 $\mathscr{C}$ 中所有的问题都可 α 图灵归约到 Π，则称 Π 是 α 图灵归约下 $\mathscr{C}$ 完全的。$\mathscr{C}$ 完全的问题是 $\mathscr{C}$ 中“最难的”问题，因为 $\mathscr{C}$ 中的问题都不比它难。设 $\mathscr{D}$ 在 α 图灵归约下是封闭的，即 $\Pi_1 \leq_t^\alpha \Pi_2$ 且 $\Pi_2 \in \mathscr{D}$ 蕴涵 $\Pi_1 \in \mathscr{D}$。那么，若 $\mathscr{C}$ 完全的问题 $\Pi \in \mathscr{D}$，则 $\mathscr{C} \subseteq \mathscr{D}$；反之，若 $\mathscr{C} \nsubseteq \mathscr{D}$，则 $\Pi \notin \mathscr{D}$。例如，多项式时间可解的判定问题类 P 在多项式时间图灵归约下是封闭的。设 Π 是多项式时间图灵归约下 $\mathscr{C}$ 完全的，那么只要 Π 是多项式时间可解的，则 $\mathscr{C}$ 中所有的问题都是多项式时间可解的；反之，如果已知 $\mathscr{C}$ 中存在非多项式时间可解的问题，则 Π 是非多项式时间可解的。于是，复杂性归约为研究计算复杂性类之间的关系和问题的复杂性下界提供了新的途径：把问题类 $\mathscr{C}$ 是否包含在 $\mathscr{D}$ 中转化为是否存在 $\mathscr{C}$ 完全的问题 Π 属于 $\mathscr{D}$；若已知 $\mathscr{C}$ 不包含在 $\mathscr{D}$ 中，只需证明 Π 是 $\mathscr{C}$ 完全的就能推出 Π 不属于 $\mathscr{D}$。

多一归约 一种常用的复杂性归约。设 Π_1 和 Π_2 是两个判定问题，f 把 Π_1 的每一个实例 I 变换成 Π_2 的实例 $f(I)$。如果对 Π_1 的每一个实例 I，I 的答案为“是”当且仅当 $f(I)$ 是 Π_2 的答案为“是”的实例，则称 f 是从 Π_1 到 Π_2 的多一归约。如果存在 Π_1 到 Π_2 的多一归约，则称 Π_1 可多一归约到 Π_2，记作 $\Pi_1 \leq_m \Pi_2$。和图灵归约类似，当限制 f 的计算复杂性为 α 时，称作 α 多一归约。当 f 为多项式时间可计算时，称作多项式时间多一归约，又称作多项式时间变换。它是 R. M. Karp 于 1972 年提出的。可以把多一归约看作图灵归约的特殊形式。设 f 是 Π_1 到 Π_2 的多一归约，如下构造 Π_1 到 Π_2 的图灵归约 A：设 S 是关于 Π_2 的假想“算法”。对 Π_1 的实例 I，先计算 $f(I)$，然后对 $f(I)$ 使用 S。S 对 $f(I)$ 的回答即为 A 对 I 的回答。因此，$\Pi_1 \leq_m \Pi_2$ 蕴涵 $\Pi_1 \leq_t \Pi_2$。在计算复杂性理论，尤其在 NP 完全性理论中，广泛使用多项式时间变换。很多完全性都是在多项式时间变换下而言的。例如，NP 完全性在多数文献中是用多项式时间变换定义的。

对数空间归约 变换 f 的空间复杂性为 $O(\log n)$ 的多一归约，又称作对数空间变换。对于对数空间归约 f 还要添加一个条件：存在多项式 p 使得，对于 Π_1 的所有实例 I，$|f(I)| \leq p(|I|)$。这里 $|I|$ 和 $|f(I)|$ 分别表示 I 和 $f(I)$ 的大小。不可能利用多项式时间变换（或任何复杂性不低于多项式时间的归约）来进一步细分 P 类。例如要考虑 DLOG 与 P 的关系，其中 DLOG 表示所有空间复杂性为 $O(\log n)$ 的判定问题。已知 DLOG⊆P，但不知道这个包含是否是真包含。为此，要定义 P 完全问题。但是，这次不再能使用多项式时间变换，而必须使用对数空间变换，因为在多项式时间变换下 DLOG 不是封闭的（除非 DLOG = P）。Π_1 可对数空间变换到 Π_2，蕴涵 Π_1 可多项式时间变换到 Π_2。也有一些文献用对数空间变换定义关于 NP 等问题类的完全性。

此外，还有 γ-归约、强非确定型多项式时间图灵归约、随机归约、真值表归约等。复杂性归约不仅可以用于判定问题，也同样可以用于函数和搜索问题。

参考文献

1. Balcázar J L, Díaz J, Gabarró J. Structural complexity Ⅰ. Berlin: Springer-Verlag, 1988
2. Balcázar J L, Díaz J, Gabarró J. Structural complexity Ⅱ. Berlin: Springer-Verlag, 1990
3. Garey M R, Johnson D S. 计算机和难解性：NP 完全性理论导引. 张立昂，沈泓，毕源章，译. 北京：科学出版社，1987

（张立昂）

fuza zhilingji jisuanji

复杂指令集计算机（complex instruction set computer，CISC） 以微程序技术为基础的具有较复杂**指令系统**的计算机。

复杂指令集计算机是相对于**精简指令集计算机**（RISC）而言的。自 20 世纪 60 年代中期 IBM360 问世以后，计算机开始走上系列化和软件向上兼容技术的主流。当时的处理机体系结构设计中是采用微程序技术作为控制指令执行的控制器的基础，即其指令系统中指令的基本操作是以微程序（又称微操作或称微指令）方式放在微存储器中。而 CISC 中的一条指令，都是由若干个微指令或微操作组成的。

如果要求该计算机产品系列的高档计算机增加功能更强、更为复杂的指令,则只需要在微存储中增加与该复杂指令相应的微程序,就可以做到计算机产品系列中从低档机到高档机的软件向上兼容。此外,以微程序技术为基础的控制器的实现是很符合当时的计算机工艺的,因为在70年代末以前,计算机的**主存储器**仍为较慢的磁心存储器;而当时的微存储器都已采用双极型半导体集成电路或其他较快的电路,主存储器的一个工作周期相当于若干个微存储器工作周期。处理机执行一条指令,至少要用一个主存储周期,相当于若干个微存储周期,即若干个微程序(微操作)周期。这样,主存储和微存储、指令和微指令是相匹配的。

从指令系统发展的道路来分析,指令系统逐渐变得复杂的原因很多,大致可归纳成三点:①在产品系列中追求软件兼容性,如Intel奔腾处理机要和Intel 80386兼容,即称为x86结构的兼容;已有的即使不合理的指令仍要保留,而新的产品又要求增加一些新的指令。②由于编写软件的工作量愈来愈大,有一个时期认为:指令系统愈复杂愈丰富,编译器愈好写,而且编译的效率愈高,就可以缓解所谓的软件危机,因此在指令系统中增加了很多接近于高级语言语句的指令,如Call/Return指令,但实际上这种指令的执行机制十分复杂。③当时主存储器价格较贵,存储器容量有限,因而往往把存储效率作为衡量处理机体系结构设计好坏的重要标准。这样,在处理机中大量采用存储效率较高的存储器-存储器操作指令;而这种指令要求微操作的数量较大,而且后来证明它是实际执行效率较低的指令类型。还有,为了提高存储效率,采用了可变字长指令字,使指令系统格式更复杂。另外,还采用各种复杂的指令寻址方式,从而使指令系统更为复杂而且非规格化。

但是,CISC与RISC实际上反映的是设计思想,具体实现时这两者的界限并不一定十分明确,如在现代RISC产品中仍设置一些复杂指令,这些复杂指令的执行控制依旧采用微程序技术,如Intel 80960 CA。此外,一些CISC处理机也采用了RISC设计思想,以减少执行一条指令所需的时钟周期数,如Intel Pentium已采用了RISC范畴的超标量结构、多流水结构和加强的编译优化技术等。

参考文献

李三立,李亚民. RISC——单发射与多发射体系结构. 北京:清华大学出版社,1994 (李三立)

fugai wangluo moshi

覆盖网络模式(overlay mode) 指将位于应用层的主机互相连接构成一个虚拟网络,由这些主机维护相互间的拓扑结构并为主机上运行的应用提供专门的服务。通常来讲,这种专门的服务比Internet提供的通用服务具有更符合用户需求的特性。使用覆盖网络模式的优势在于不需要部署新的网络设备,也不必修改现有的网络协议和软件,只需在现有网络协议之上部署支持自身应用的软件协议即可。

在覆盖网络模式中,主机之间通过虚拟链路/逻辑链路互联而成的网络称为覆盖网络,又称应用层网络。它构建于现有的物理网络之上,是一个完全位于应用层的网络系统。

使用覆盖网络模式可以提高路由的可靠性,典型代表是MIT提出的**弹性覆盖网络**(RON)。RON是一种分布式覆盖网络体系结构,可以使分布式应用检测到路径的失效和周期性的性能降低现象并能够从中迅速恢复(恢复时间少于20s),而目前Internet中使用的BGP协议通常需要几分钟的时间才能从失效中恢复。

覆盖网络模式还广泛应用于实现组播服务。目前Internet的网络传输还不能全面支持组播,而多媒体应用的发展又迫切需要组播功能的支持,可以将参加组播的计算机构成一个覆盖网络,提供应用层组播。

应用层组播(application layer multicast)是利用构建在应用层之上的覆盖网络实现组播通信的技术,其主要思想为:参与组播通信的端节点利用端到端链接形成一个应用层的覆盖网络,然后由组播路由协议在该覆盖网络上构建组播转发树,完成数据转发。应用层组播构建在应用层之上,利用端节点间的单播链接完成组播通信,取消了对网络中间节点(路由器)的特殊支持的要求,因此得到了快速的部署和发展。应用层组播主要包含两方面的内容,即覆盖网络的建立与维护、路由算法设计。

当前的覆盖网络可以分为两类,树形结构和网状结构。与IP组播由路由器构建组播树的概念相仿,基于树形结构的应用层组播利用端节点直接构建起一个应用层的组播树。新参与该组播通信的节点显式地从已有节点中选择父节点,父节点负责向其所有子节点转发数据。树形结构的主要优点是结构简单、易于维护、控制开销小,而其主要缺点有父节点失效会影响其所有子节点、系统稳定性差、带宽利用不均、大量叶节点上传带宽被浪费。与树形结

构直接构建组播树不同,网状结构先建立一个节点间相互连接的网络拓扑,再在这个拓扑上建立由源节点到接收节点的组播树。

路由算法的设计也即组播树的构建,这可以解释为如下的一个图问题:给定一张图及图上各节点的限制,构建满足一定条件的连接所有节点的子结构。这一方面受限于覆盖网络的选择;一方面也受各节点出、入带宽的限制;最后,还有诸如最小全局开销、最小最大延迟等最终目标的影响。通常的路由算法有以下几类:

(1) 最短路径算法　这类算法考虑节点度数受限情况下的最小直径生成树问题。通过构建由从源点到所有接收节点的最短路径组成的生成树,减小系统的最大延迟。这属于源树(source specific tree)。

(2) 最小生成树算法　组播树为一个连接各节点的最小生成树,以实现全局的最小开销。这属于共享树(share tree)。

(3) 分簇算法　这类算法以层次化的簇结构组织组播树。一定数目的节点组成一个簇,各簇中选择一个头节点组成上一层的节点集合,源节点位于最高层,数据由上向下转发。

(4) P2P 算法　这类算法主要用于利用已有 P2P 机制构建覆盖网络的应用层组播协议中。组播树的构建通常采用逆向/前向路径转发机制实现。

参考文献

1. Andersen D, Balakrishnan H, Kaashoek M, Morris R. Resilient overlay networks. In: Proc 18th ACM SOSP, Oct. 2001

2. Jannotti J, Gifford DK, Johnson KL, et al. Overcast: reliable multicasting with an overlay network. In: Proc USENIX OSDI, Oct. 2000

3. Hosseini M, Ahmed DT, Shirmohammadi S, et al. A survey of application-layer multicast protocols. IEEE Communications Surveys and Tutorials, 2007

(徐恪　陈贵海)

G

gailütu moxing

概率图模型(probabilistic graphical models)

一种结构学习的方法,它用图表示随机变量之间条件独立性的概率模型(参见**机器学习**)。是一种描述复杂随机系统的有效框架。

概率图模型的研究最早可以追溯到20世纪初期统计物理学家使用无向图来表示相互作用的粒子系统的概率分布,但是直到20世纪80年代后期才被**人工智能**、**机器学习**等领域广泛地接受和发展。目前,概率图模型已被广泛应用于**计算机视觉**、**自然语言处理**、计算生物学等诸多领域。

概率图模型一般由表示、推理、学习等三部分内容组成。

表示:概率图模型使用图表示随机变量之间的依赖关系,其中图上的结点表示描述问题的随机变量,边表示随机变量之间的直接依赖关系。概率图模型主要分为有向图模型(也称**贝叶斯网**)和无向图模型(也称马尔可夫网络或马尔可夫随机场),如图1所示。它们分别描述了随机变量 A、B、C 和 D 的概率依赖关系。

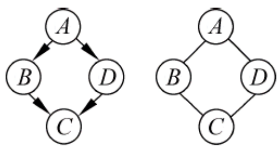


图1 包含四个随机变量的有向图模型(左)和无向图模型(右)

两类模型的区别在于它们的语义和概率描述不同。其中语义是指条件独立性断言的集合,而概率描述是指联合概率分布如何由局部因子表示。对于同一个概率图模型,语义和概率描述是等价的两个方面。有向图模型的条件独立性断言为:给定父结点变量,一个随机变量和它的非后代结点变量条件独立;它的概率描述为联合概率分布,可以表示为一组条件概率分布的乘积

$$p(X_1,\cdots,X_n) = \prod_{i=1}^{n} p(X_i \mid Pa_i),$$

其中 Pa_i 表示变量 X_i 的父结点变量。无向图模型的条件独立性断言为:给定一个随机变量的邻居结点变量,该变量和其他非邻居结点变量独立;它的概率描述也是一个联合概率分布,其形式为

$$p(X_1,\cdots,X_n) = \frac{1}{Z}\prod_{c\in C}\phi_c(X_1,\cdots,X_n),$$

其中 C 表示该图的团图的集合,ϕ_c 是定义在和团图 c 相关的随机变量上的因子,Z 是归一化因子,也称配分函数。这种形式的分布被称为吉布斯分布。

推理:从给定概率图模型中推导查询任务的答案。可能的查询任务包括:计算部分随机变量的边缘概率、条件概率或期望值等。对于树宽较小的图模型,可以使用精确算法,比如连接树算法等有效地进行推理。而对于树宽较大的一般图模型,精确算法通常不可行。在这种情况下,需要使用近似算法在有限时间内获得一定程度近似好的解。常用的近似算法分为两大类——采样法和变分法。采样法的基本原理是通过从某个概率分布随机抽取一定数目的样本来估计要推理的目标;而变分法的一般思想是根据优化理论将推理变成求解某个优化问题,并且通常对优化问题引入适当的约束以推导出有效的近似推理算法。这两种推理方法互有优缺点。比如采样法的优点是实现简单,但是它通常收敛比较慢而且比较难以检查是否收敛;变分法的优点是它显式地求解一个确定的优化问题,其中优化的目标函数通常是要求解目标的上界或者下界,而变分法的缺点是在有些情况下不如采样法准确。为了权衡近似精度和计算效率,这两种推理方法都需要有效地利用由模型的图结构所表示的变量之间的条件独立性。在实际应用中,这两种方法并不冲突,有时候将二者综合使用可以得到比使用任何一种方法都更好的解决方案。目前,推理算法的研究热点主要有三个方面:一是研究通用的、高近似精度的推理算法;二是研究特定领域或特定模型的高近似精度的推理算法;三是研究大规模应用场景下的近似推理算法,如分布式推理算法等。

学习:分为参数学习和结构学习。参数学习是指在给定图结构的情况下从训练数据中学习联合概率分布的参数,比如贝叶斯网络中的条件概率分布

等。结构学习是指从训练数据中学习模型的图结构。相比较而言,结构学习比参数学习更难,通常参数学习是结构学习的一个子步骤。由于通用算法的存在,在提供粗略的领域知识(比如部分主要随机变量及它们之间的依赖关系等)的前提下,人们可以方便地学习一个概率图模型(可能包括新的随机变量及新的依赖关系等)以有效地近似描述复杂的实际系统。相比于手工设计,结构学习通常可以获得更好反映真实世界的图模型,并且对于发现复杂系统中随机变量之间隐含的深层依赖关系等具有重要的理论和实际意义。目前,结构学习的研究主要有两个方面:一是研究准确、有效的结构学习算法;二是研究结构学习在深入理解科学及应用问题方面的应用。

参考文献

1. Koller D, Friedman N. Probabilistic graphical models: Principles and techniques. Cambridge, MA: MIT Press, 2009

2. Pearl J. Probabilistic reasoning in intelligent systems: Networks of plausible inference. Morgan Kaufmann Publishers Inc., 1988

3. Wainwright M J, Jordan M I. Graphical models, exponential families, and variational inference. Foundations and Trends in Machine Learning, 2008, 1(1-2): 1-305

(朱军)

gailü zidongji

概率自动机(probabilistic automaton) 一种其结构和所处环境具有随机因素的自动机。又称随机自动机。对于概率自动机,要给出它的初始分布,即开始时内部状态的概率分布;还要规定在某种条件下,概率自动机下一个动作的条件概率。通常,概率自动机的执行结果随其结构不同给出某种概率。概率自动机在信息论、容错计算、学习模型和控制模型等方面都有广泛应用。

概率自动机包括概率时序机、概率有限自动机、概率下推自动机和概率图灵机,主要研究它们的功能、结构、化简和其他性质。概率文法从语言的角度给出了另一研究方向。

概率时序机 时序机的一种推广,它刻画的功能是在给定的初始分布 π 下,输入符号序列 u 后产生输出符号序列 v 的概率 $P_\pi(v|u)$。形式上

$$M=(X,Y,Q,P,\pi)$$

其中 X 和 Y 分别是有限的输入和输出符号集;$Q=\{q_1,q_2,\cdots,q_n\}$ 是有限状态集;$\pi=[\pi_1\pi_2\cdots\pi_n]$ 是初始分布,π_i 表示开始时 M 处于状态 q_i 的概率,满足 $\pi_i\geqslant 0, 1\leqslant i\leqslant n$,且 $\sum_{i=1}^{n}\pi_i=1$;P 是条件概率

$$P(y,q'\mid q,x);\ x\in X;\ y\in Y;\ q,q'\in Q$$

表示 M 处于状态 q,输入符号 x 时,输出符号为 y 且下一状态为 q' 的概率,满足 $P(y,q'\mid q,x)\geqslant 0$ 且 $\sum_{y\in Y}\sum_{q'\in Q}P(y,q'\mid q,x)=1$。设 $P(y\mid x)$ 表示以 $P(y,q_j\mid q_i,x)$ 为元素的 $n\times n$ 矩阵,η 是各分量都是 1 的 n 维列向量,则 $P_\pi(y_1y_2\cdots y_k\mid x_1x_2\cdots x_k)=\pi P(y_1\mid x_1)P(y_2\mid x_2)\cdots P(y_k\mid x_k)\eta$。例如,设 $X=\{0,1\}, Y=\{a,b\}, Q=\{q_1,q_2\}, \pi=\begin{bmatrix}\frac{1}{4} & \frac{3}{4}\end{bmatrix}$,且 P:

$$P(a\mid 0)=\begin{bmatrix}\frac{1}{2} & 0\\ 0 & \frac{1}{2}\end{bmatrix},\quad P(b\mid 0)=\begin{bmatrix}\frac{1}{4} & \frac{1}{4}\\ \frac{1}{2} & 0\end{bmatrix}$$

$$P(a\mid 1)=\begin{bmatrix}0 & \frac{1}{2}\\ 0 & 0\end{bmatrix},\quad P(b\mid 1)=\begin{bmatrix}\frac{1}{2} & 0\\ \frac{1}{2} & \frac{1}{2}\end{bmatrix}$$

则 $P_\pi(a\mid 0)=\pi P(a\mid 0)\eta=\begin{bmatrix}\frac{1}{4} & \frac{3}{4}\end{bmatrix}\times\begin{bmatrix}\frac{1}{2} & 0\\ 0 & \frac{1}{2}\end{bmatrix}\begin{bmatrix}1\\1\end{bmatrix}=\frac{1}{2}$, $P_\pi(ab\mid 00)=\pi P(a\mid 0)$

$$P(b|0)\eta=\begin{bmatrix}\frac{1}{4} & \frac{3}{4}\end{bmatrix}\begin{bmatrix}\frac{1}{2} & 0\\ 0 & \frac{1}{2}\end{bmatrix}\begin{bmatrix}\frac{1}{4} & \frac{1}{4}\\ \frac{1}{2} & 0\end{bmatrix}\begin{bmatrix}1\\1\end{bmatrix}=\frac{1}{4}$$

设 π 和 π' 分别为 $\pi_i=1$ 和 $\pi_j=1$ 的两个初始分布。如果对于任意的正整数 m 和所有的 $x_k\in X$, $y_k\in Y, 1\leqslant k\leqslant m$, $P_\pi(y_1y_2\cdots y_m\mid x_1x_2\cdots x_m)=P_{\pi'}(y_1y_2\cdots y_m|x_1x_2\cdots x_m)$ 总成立,则称状态 q_i 和 q_j 等价。利用状态等价的概念,可以化简概率时序机。研究状态等价的充分必要条件,寻求给定概率时序机的化简形式和化简方法,是重要的研究内容。

概率有限自动机 有限自动机的推广,它刻画的功能是在给定的初始分布 π 下,输入符号序列 u 后达到给定状态集 F 中的状态的概率 $P_\pi(u)$。形式上

$$M=(X,Q,P,\pi,F)$$

其中 X 和 $Q=\{q_1,q_2,\cdots,q_n\}$ 分别是有限输入符号

集和状态集；π 是初始分布；F 是给定的状态集，$F \subseteq Q$；P 是条件概率

$$P(q'|q,x),\quad x \in X;\ q,\ q' \in Q$$

表示 M 处于状态 q，输入符号 x 时，下一状态为 q' 的概率，满足 $P(q'|q,x) \geqslant 0$ 且 $\sum_{q' \in Q} P(q'|q,x) = 1$。如果 η^F 是一 n 维列向量，它相应于 F 中状态的分量为1，其余分量为0，则 M 在初始分布 π 下，输入符号序列 $x_1x_2\cdots x_m$ 后达到 F 中状态的概率 $P_\pi(x_1x_2\cdots x_m) = \pi P(x_1)P(x_2)\cdots P(x_m)\eta^F$，其中 $P(x)$ 表示以 $P(q_j|q_i,x)$ 为元素的 $n \times n$ 矩阵。给定一个阈值 λ，$0 \leqslant \lambda < 1$，则 M 以切断点 λ 所识别的随机语言

$$T(M,\lambda) = \{u: u \in X^*, P_\pi(u) > \lambda\}$$

通常，$P(q'|q,x) < 1$。于是，随着输入符号序列 u 的长度增加，$P_\pi(u)$ 趋于0。这样，$\lambda = 0$ 就成为人们最感兴趣的切断点。当 π_i，$1 \leqslant i \leqslant n$，和 $P(q'|q,x)$ 仅取值1或0时，概率有限自动机蜕化为有限自动机，所以随机语言类包含正则语言类。随机语言类的结构，对运算的封闭性，概率有限自动机的各种模型及其相互关系，都是重要的研究课题。

概率下推自动机 下推自动机的一种推广，它刻画的功能是在给定的状态和下推符号对 (q,Z) 的初始分布 $h(q,Z)$ 下，输入符号序列 u 后产生输出序列 v 且达到状态 q' 的概率。形式上

$$M = (X,Y,W,Q,P,h,g)$$

其中 X，Y 和 Q 分别是有限的输入、输出符号集和状态集；$h = h(q,Z)$ 是 $Q \times W$ 到 $[0,1]$ 内的函数，满足 $\sum_{q \in Q}\sum_{Z \in W} h(q,Z) \leqslant 1$；$g = g(q)$ 是 Q 到 $[0,1]$ 内的函数；P 是条件概率

$$P(q',z,v|x,Z,q),\quad q,q' \in Q; x \in X; Z \in W; z \in W^*; v \in Y^*$$

表示 M 处于状态 q，栈顶符号为 Z，输入符号为 x 时，下一状态为 q'，栈顶符号改写为 z 且输出为 v 的概率，满足 $P(q',z,v|x,Z,q) \geqslant 0$，$\sum_{q' \in Q}\sum_{z \in W^*}\sum_{v \in Y^*} P(q',z,v|x,Z,q) \leqslant 1$ 且 $\{P: P(q',z,v|x,Z,q) \neq 0, z \in W^*, v \in Y^*\}$ 为有限集。概率下推自动机简记为PPA。设 $\Gamma^M = Q \times W^* \times X^* \times Y^*$，定义 Γ^M 到 Γ^M 的随机函数 $P_k^M(\beta|\alpha)$：

$$P_1^M(\beta|\alpha) = \begin{cases} P(q',z_0,v_0|x,Z,q), & \text{如果 } \alpha = (q,Zz,xu,v), \\ & \text{且 } \beta = (q',z_0z,u,vv_0) \\ 0, & \text{否则} \end{cases}$$

$$P_{k+1}^M(\beta|\alpha) = \sum_{\gamma \in \Gamma M} P_k^M(\beta|\gamma)P_1^M(\gamma|\alpha),\quad k \geqslant 1$$

这样，输入符号序列 u 后产生输出符号序列 v 的概率是 $G^M(v|u) = \sum_{q,q' \in Q}\sum_{Z \in W}\sum_{z \in W^*} h(q,Z) g(q')P_k^M(q',z,\varepsilon,v|q,Z,u,\varepsilon)$，其中 $k = \lg(u)$ 是 u 的长度。

如果PPA $M = (X,Y,W,Q,P,h,g)$ 满足(1) 对于任意的 $x \in X, Z \in W$ 和 $q \in Q$，$\sum_{q' \in Q}\sum_{z \in W^*}\sum_{v \in Y^*} P(q',z,v|x,Z,q) = 1$，(2) $\sum_{q \in Q} \times \sum_{Z \in W} h(q,Z) = 1$ 和(3) 对于任意的 $\alpha = (q,Z,u,\varepsilon)$，$q \in Q, Z \in W, u \in X^*$，$\sum_{\beta \in \Gamma M} P_k^M(\beta|\alpha) = 1$，其中 $k = \lg(u)$，则称 M 为完全的PPA。可以证明，对于任意的PPA M，存在完全的PPA M'，使得 $G^M = G^{M'}$。

PPA M 接受的语言为 φ^M：$\varphi^M(u) = \sum_{v \in Y^*} G^M(v|u)$。如果 f 是PPA M 接受的语言，则有完全的PPA $M' = (X,Y,W,Q,P,h,g)$ 接受 f：$f = \varphi^{M'}$，满足 $Y = \{y\}$ 且 $P(q',z,v|x,Z,q) \neq 0$ 隐含 $v = y$。这时，M 可以表示为一个六元组 $A = (X,W,Q,P',h,g)$，其中 $P'(q',z|x,Z,q) = P(q',z,y|x,Z,q)$。$A$ 称为概率下推接受器。还可以定义带有切断点 λ 的PPA M 接受的语言 $L(\varphi^M,\lambda) = \{u: u \in X^*, \varphi^M(u) > \lambda\}$，$0 \leqslant \lambda < 1$。带切断点 λ 的PPA接受的语言类真包含随机语言类和上下文无关语言类。

概率下推自动机理论研究PPA的各种限制类型和接受的语言类，以及这些语言类对运算的封闭性和相互关系等问题。

概率图灵机 图灵机的推广。形式上

$$M = (W,Q,P,\pi)$$

其中 Q 为有限状态集；W 是带符号集；π 是初始分布；P 是条件概率

$$P(y,q'|q,x),\quad q,q' \in Q; x \in W; y \in V, \quad V = W \cup \{R,L,T\}$$

表示 M 处于状态 q，读写头注视带符号 x 时，M 的"下一动作"的概率，满足 $P(y,q'|q,x) \geqslant 0$ 且 $\sum_{y \in V}\sum_{q' \in Q} P(y,q'|q,x) = 1$。"下一动作"为下列四者之一：①$y = R$：读写头右移一格，状态转移到 q'；②$y = L$：读写头左移一格，状态转移到 q'；③$y \in W$：读写头不动，将注视的带符号 x 改写为 y；④ $y = T$：读写头不动，机器停止。

用概率图灵机可以定义可计算随机函数和带有

阈值 λ 的可计算随机函数。可计算随机函数类对函数代入、原始递归、求极小等运算都是封闭的。限制在普通函数类的范围内,部分可计算随机函数中的普通函数,恰好是部分递归函数。带有阈值 λ 的可计算随机函数类是不可数的。

概率文法 短语结构文法的推广,它刻画的功能是给定生成式的初始分布和确定生成式先后使用的概率后,短语结构文法生成的终结符号序列的概率。形式上

$$G_P = (G,P,\pi)$$

其中 $G = (N,\sum,\{f_1,f_2,\cdots,f_n\},X_0)$ 是短语结构文法;$\pi = [\pi_1\ \pi_2 \cdots\ \pi_n]$ 是生成式的初始分布;P 是条件概率

$$P(f_k \mid f_j),\quad 1 \leqslant j,k \leqslant n$$

表示 G_P 使用生成式 f_j 后,下一个使用的生成式为 f_k 的概率,满足 $P(f_k|f_j) \geqslant 0$ 且 $\sum_{k=1}^{n} P(f_k|f_j) = 1$。如果 G 是 i 型文法,则称 G_P 是 i 型概率文法,$i=0,1,2,3$。设 G_P 的派生

$$D:\ X_0 = u_0 \Rightarrow_{f_{j(1)}} u_1 \Rightarrow_{f_{j(2)}} u_2 \Rightarrow \cdots \Rightarrow_{f_{j(r)}} u_r,$$
$$u_r \in \sum{}^*$$

则派生 D 的概率 $P(D) = \pi_{j(1)} P(f_{j(2)} \mid f_{j(1)}) \cdots P(f_j(r) \mid f_{j(r-1)})$。如果给定阈值 λ,$0 \leqslant \lambda < 1$,则带有切断点 λ 的概率文法 G_P 生成的语言 $L_s(G_P,\lambda) = \left\{u\colon\ u \in \sum{}^*,\ \sum_D P(D) > \lambda, D \text{ 为生成 } u \text{ 的派生}\right\}$,和 $L_m(G_P,\lambda) = \{u\colon\ u \in \sum{}^*$,存在生成 u 的派生 D,使得 $P(D) > \lambda\}$。例如,2 型概率文法 $G_P = (\{X_0,X\},\{a,b,c\},\{f_1,f_2,f_3,f_4,f_5\},X_0,P,[1\ 0\ 0\ 0\ 0])$,其中 f_j,$j=1,2,3,4,5$,依次是 $X_0 \to X_0X$, $X_0 \to aX_0b$, $X \to cX$, $X_0 \to ab$, $X \to c$;P:$[P(f_1|f_j)P(f_2|f_j)P(f_3|f_j)P(f_4|f_j)P(f_5|f_j)]$,$j=1,2,3,4,5$,依次是 $[0\ 1/2\ 0\ 1/2\ 0]$,$[0\ 0\ 1\ 0\ 0]$,$[0\ 1/2\ 0\ 1/2\ 0]$,$[0\ 0\ 0\ 0\ 1]$,$[0\ 0\ 0\ 0\ 1]$。则 $L_s(G_P,0) = L_m(G_P,0) = \{a^l b^l c^l\colon\ l \geqslant 1\}$。

如果记 $P \cdot i \cdot S = \{L_s(G_p,\lambda)\colon\ G_P$ 是 i 型概率文法,$0 \leqslant \lambda < 1\}$ 和 $P \cdot i \cdot m = \{L_m(G_P,\lambda)\colon\ G_P$ 是 i 型概率文法,$0 \leqslant \lambda < 1\}$,则有 $P \cdot 3 \cdot S = P \cdot 3 \cdot m =$ 正规语言类。在 i 型概率文法 $G_P = (G,P,\pi)$ 中,π 代之以具有非负分量的 n 维向量 δ,P 代之以 φ:$\varphi(f_k|f_j) \geqslant 0$,$1 \leqslant j,k \leqslant n$,则称 $G_W = (G,\varphi,\delta)$ 为 i 型加权文法。可以证明,随机语言类 $P \cdot T = \{T(M,\lambda)\colon$ M 是概率有限自动机,$0 \leqslant \lambda < 1\} \subseteq \{L_s(G_W,\eta)\colon\ G_W$ 是 3 型加权文法,$\eta \geqslant 0\}$。反之,$\{L_s(G_W,\eta)\colon\ G_W$ 是 3 型加权文法,且不含形如 $X \to Y$ 的生成式,X 和 Y 是变量;$\eta \geqslant 0\} \subseteq P \cdot T$。研究概率文法和加权文法的层次结构及生成的语言类之间的相互关系,各语言类对运算的封闭性及判定问题等都是重要的研究方向。

参考文献

1. Paz A. Introduction to probabilistic automata. New York: Academic Press, 1971

2. Santos E S. Probabilistic pushdown automata. Journal of Cybernetics. 1976, 6: 173-187

3. Salomaa A. Probabilistic and weighted grammars. Information and Control. 1969, 15: 529-544

(郭清泉)

ganzhiji

感知机(perceptron) 一种模仿人的记忆和学习机制的简单**人工神经网络**模型,是美国学者 F. Rosenblatt 于 20 世纪 50 年代末提出的,具体结构由 S(传感单元)层,A(联想单元)层和 R(响应单元或输出单元)层共三层神经元组成(见图 1)。只有 A 层与 R 层间的连接权值 w_i 是可以学习的,可见它是只有一层计算单元(R 层)的神经网络;而且同一层内的神经元之间没有相互连接,不同层之间也没有反馈,所以实际上是一个单层前向神经网络。

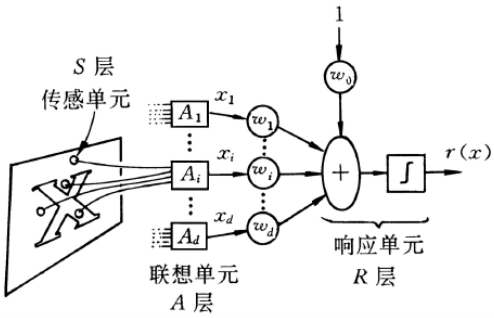


图 1 感知机的基本构成

虽然联想单元的个数可以很多,通常它仍远小于传感单元个数,每个联想单元的输出是传感单元信号的某一固定的线性组合,分别记为 $x_1,x_2,\cdots,x_d$,它们经可调权值 $w_1,w_2,\cdots,w_d$ 送到响应单元 R,先经线性组合成为

$$w_0 + \sum_{i=1}^{d} w_i x_i = w_0 + W^{\mathrm{T}} X$$

再经取阈值,最后输出为

$$r(x) = \begin{cases} 1, & w_0 + W^{\mathrm{T}} X \geqslant 0 \\ 0, & w_0 + W^{\mathrm{T}} X < 0 \end{cases}$$

根据 $r(x)$ 的取值可将输入样本分类,因此它是一个线性分类器。

在实际应用中,要先用已知类别的样本(训练集)对网络训练以确定权值 W,具体过程如下:

设在 t 时刻输入训练样本 $X_s=(x_{s1},x_{s2},\cdots,x_{sd})$,此时刻网络输出为 $r_s(x)$,而相应 X_s 的正确输出为 y_s,记 $\varepsilon_s=y_s-r_s(x)$ 为网络实际输出与正确输出的误差,则 w_i 按下式修正:

$$w_i(t+1)=w_i(t)+\varepsilon_s x_{si}=w_i(t)+[y_s-r_s(x)]x_{si}$$

以上学习算法称为感知机学习算法,可以证明,当输入样本是线性可分时,该算法可以在有限步骤内收敛,且训练后的网络可对其他输入样本正确分类。也可以用均方误差达最小的学习算法,当输入不是线性可分时它收敛于使均方误差最小的解。

感知机的输出所形成的函数是一个 d 维空间中的超平面,所以原则上它只能解决线性可分的模式分类问题。对于非线性可分的模式,用感知器学习算法时不论怎么调整权值,都不能实现正确分类。一个简单的例子是两个输入一个输出的异或运算[输入为(0,1)或(1,0)时对应输出应为 1,输入(0,0)或(1,1)时输出应为 0]问题,感知机就不能胜任。对于感知机的能力及其局限性,Minsky 和 Papert 在他们的著作"Perceptron"中有详细分析。

虽然感知机在实用中有很大局限性,但是 F. Rosenblatt 通过感知机模型的研究首次提出了神经网络模型的学习概念并给出了一种有效的学习算法,对神经网络的发展起了很大作用。

参考文献

1. 史忠植. 神经网络. 北京:高等教育出版社,2009

2. Васцеь В И. 机器识别方法与系统. 边肇祺,阎平凡,译. 北京:科学出版社,1991

3. Minsky M, Papert S. Perceptron. MIT Press, 1988

(阎平凡)

gaoceng tixi jiegou

高层体系结构(high level architecture, HLA) 一个实现联邦式协同仿真的协议。它具有开放的体系结构、能提供将多种异构仿真实体集成在一起的接口规范,实现各类仿真应用间的互操作、系统和部件的重用以及建立不同层次和不同粒度的对象模型,并能够根据不同的用户需求和不同的应用目的,实现协同仿真联邦的快速组合和重新配置的分布式仿真。

高层体系结构是分布交互式仿真(参见**分布交互式仿真**)的高级阶段。分布交互式仿真起于美国国防部高级研究计划局(DARPA)和美国陆军制定的 SIMNET 研究计划,经历 SIMNET 研制和使用、DIS 研制和使用、ALSP 阶段的发展后,人们发现,SIMNET、DIS、ALSP 都是同类功能仿真应用(武器平台、模拟仿真器、计算机生成兵力 CGF、聚合级仿真模型)的互连,只有有限的互操作性,不能满足越来越复杂的作战仿真需求。为此,美国国防部 1995 年发布的建模与仿真主计划(MSMP),决定在吸收和继承 DIS 与 ALSP 协议有益思想的基础上,在国防部范围内建立一个通用的仿真技术框架来保证国防部范围内的各种仿真应用之间的互操作性,技术框架的核心是高层体系结构(HLA)。

建模与仿真主计划提出了未来建模/仿真的共同技术框架。它包括三个方面:高层体系结构(HLA)、任务空间概念模型(CMMS)和数据标准(DS)。它们的共同目标是实现仿真间的互操作,并促进仿真资源的重用。具体地说,就是通过计算机网络使得分散分布的各仿真部件能够在一个统一的仿真时间和仿真环境下协调运行,且可以重复使用。HLA 的基本思想就是使用面向对象的方法,设计、开发及实现系统不同层次和粒度的对象模型,来获得仿真部件和仿真系统高层次上的互操作性与可重用性。在 2000 年 9 月 22 日由美国 IEEE 标准化委员会正式定为 IEEE1516 标准,随后 OMG、北约 M&S 组织也采纳 HLA 作为标准。IEEE1516 标准的确定,标志着分布交互式仿真技术发展到了成熟阶段。

HLA 要点说明 HLA 主要由三个部分组成:HLA 规则(HLA rules)、HLA 对象模型模板(object model template, OMT)和联邦接口规范(federate interface specification)。

(1) HLA 规则 HLA 规则定义了 HLA 的设计目标以及对 HLA 兼容的联邦成员和联邦的约束;保证联邦(federation)中仿真应用间按正确的方式进行交互,描述各联邦成员的责任及它们与运行支撑环境(run time infrastructure, RTI)的关系。

(2) HLA 对象模型模板 HLA 对象模型模板(OMT)规定了描述 HLA 对象模型信息的通用方法,提供一种标准格式的模型模板,以促进模型的互操作性和资源的可重用性。

OMT 规定了所有联邦对象模型(FOM)的结构,每一个联邦均有对应的 FOM。FOM 中只定义联邦成员间共享信息,不描述联邦成员内部的事情。

FOM 是联邦运行时通过运行支撑环境 RTI 交换数据的唯一词汇表。在联邦运行的开始,FOM 作为数据提供给 RTI。当 FOM 改变时 RTI 并不需要改变,这是 HLA 的一个重要特征,这意味着 HLA 实现了模型数据驱动仿真实验,而实验环境 RTI 可以适应不同的仿真模型,只要通过 OMT 定义的 FOM 即可。

(3) HLA 接口规范(interface specification) HLA 接口规范包含 RTI 能向联邦成员提供的服务以及联邦成员能为 RTI 所提供的服务,利用这些服务,联邦成员在参与联邦执行时能够与其他成员有效地进行信息交换。

上述三者的关系是:规则集中规定了联邦对象模型 FOM 和仿真对象模型 SOM 的建立必须用 OMT 记录,OMT 保证联邦成员与 RTI 进行方便的交互,三者相互作用,紧密相关,任何一部分的改动都会影响其他部分的改变。因此,三者统称为 HLA 标准,是随时间推进而不断演变的。例如,目前 IEEE1516 标准有:①IEEE 1516—2000-HLA 建模与仿真框架与规则标准,包含 IEEE 1516. 1—2000-HLA 建模与仿真联邦接口描述标准,IEEE 1516. 2—2000-HLA 建模与仿真对象模型模板描述标准,IEEE 1516. 3—2003-HLA 联邦开发与执行过程推荐实施标准,IEEE 1516. 4—2007-HLA 联邦开发与执行过程中 VV&A 推荐实施标准。②IEEE 1516—2010-HLA 建模与仿真框架与规则标准,包含 IEEE 1516. 1—2010-HLA 建模与仿真联邦接口描述标准,IEEE 1516. 2—2010-HLA 建模与仿真对象模型模板描述标准。

运行支撑环境 RTI 运行支撑环境 RTI 是 HLA 仿真系统进行分层管理控制、实现分布仿真可扩充性的支撑基础。运行支撑环境 RTI 是按照 HLA 开发的服务程序,它实现了 HLA 接口规范中所有的服务功能。

如果说 HLA 是建模与仿真的技术框架的核心,那么 RTI 则是核心中的核心,建立联邦成员必须以 RTI 为其支撑软件,仿真过程中需要交互的各种信息都要通过 RTI 来完成。

HLA 是通过 RTI 来实现的,它将应用层同其底层支撑环境功能分离,即将具体的仿真功能实现、仿真运行管理和底层传输三者分离开来,从而使各部分的开发与实现相对独立。

RTI 提供 6 大类服务,即联邦管理、声明管理、对象管理、所有权管理、时间管理和数据分发管理等。从联邦成员看 RTI,联邦成员的变化不会影响 RTI,不同的联邦成员可使用同一个 RTI,只要它们遵循相关规则和规范。从 RTI 看联邦成员,RTI 为联邦成员提供接口,尽管联邦成员的所有状态被屏蔽,同时联邦成员也为 RTI 提供接口。RTI 有多个联邦成员,每一个都有同样的接口,只是标识不同而已。

RTI 提供给每一个联邦成员的接口称为 RTI 大使(RTI ambassador),联邦成员调用该接口的操作请求 RTI 服务,每一个联邦成员也对 RTI 提供一个称为联邦成员大使(federate ambassador)的接口,当 RTI 必须通知联邦成员时,RTI 通过调用联邦成员大使的操作来实现。基于 HLA 协同仿真系统体系结构如图 1 所示。

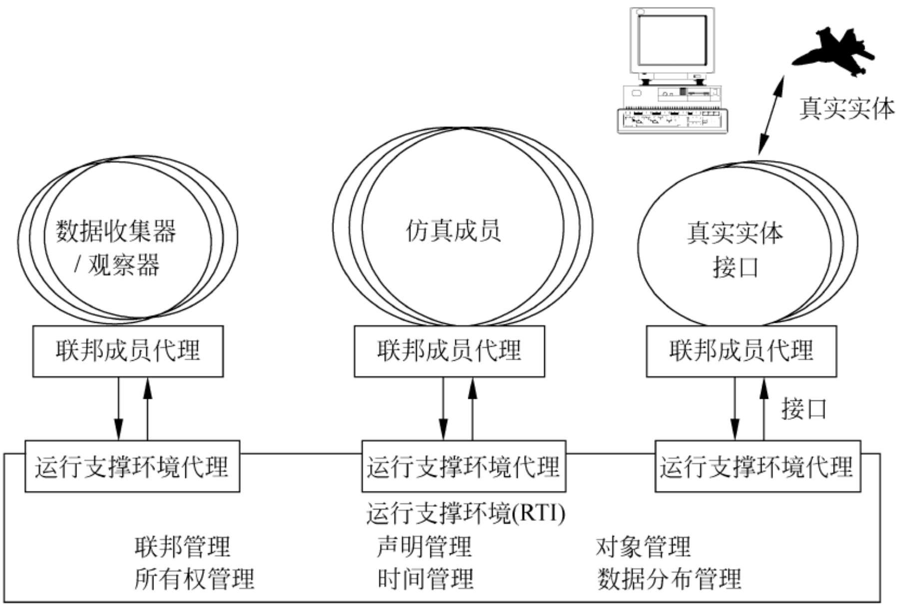


图 1 基于 HLA 的协同仿真系统体系结构

HLA 的主要目标是允许通过联合各种仿真系统来实现仿真的应用,因此支持基于组件的仿真开发。HLA 是一种支持基于组件的仿真体系结构。组件可以理解为联邦成员,而联邦成员是一个仿真系统或工具。联邦由联邦成员构成,是软件重用的基本单元。联邦成员是完整的运行子系统,而不是库里的例程或对象。对象是指对某一应用领域内所要进行仿真的实体建立的模型。

联邦执行过程称为联邦执行;在 HLA 框架下,联邦成员通过 RTI 构成一个开放性的分布式仿真系统,在联邦的运行阶段,这些成员之间的数据交换必须通过 RTI。

HLA 是目前流行的分布仿真技术。自 2000 年被确立为 IEEE1516 标准至今,其在世界军事仿真领域得到了广泛的应用。HLA 提出的最初目的是提高仿真组件的可重用性和互操作性,随着应用的不断深入,这一目标在一定程度上得到了实现。但是目前的建模与仿真系统依旧面临着一系列问题和挑战;例如:HLA 运行支撑环境 RTI 的实现与特定编程语言及计算机平台有关,互操作性有限,需要引入新的技术以实现仿真应用在更大规模上的互操作性;与其他领域的许多标准脱节,特别是 Web 技术并未得到应用等。为解决这些问题,建模与仿真需要确定和采纳有关的新技术。

目前已经出现的有可扩展建模与仿真框架 XMSF (eXtensible modeling and simulation framework)。它使用各种开放标准(Web 服务、可扩展标记语言 XML 等),定义了一组基于 Web 的建模/仿真标准、描述以及推荐准则等,为未来的建模与仿真应用创造一个可扩展的框架,以促进建模与仿真应用在更大范围的互操作性和重用性。

参考文献

1. 柏彦奇. 联邦式作战仿真. 长沙:国防科技大学出版社,2001

2. 周彦,戴剑伟. HLA 仿真程序设计. 北京:电子工业出版社,2002

3. 郭齐胜. 分布交互仿真及其军事应用. 北京:国防工业出版社,2003

(范文慧　肖田元)

gaoci daishu fangcheng shuzhi jiefa

高次代数方程数值解法(numerical solution of polynomial equation) 采用数值计算方法求 n 次多项式方程

$$P_n(x) \equiv a_0x^n + a_1x^{n-1} + \cdots + a_{n-1}x + a_n = 0 \quad (1)$$

的根(即 $P_n(x)$ 的零点)的方法。式(1)中,a_0, a_1, …, a_n 为实数或复数,且 $a_0 \neq 0$, $n \geq 2$ 为正整数。由代数方程基本定理,在复数域内 n 次代数方程有 n 个根 x_1, x_2, …, x_n,其中可能有相同的根,称为重根。对二次方程 $a_0x^2 + a_1x + a_2 = 0$, $a_0 \neq 0$,它的两个根可以表示为

$$x_1, x_2 = \frac{-a_1 \pm \sqrt{a_1^2 - 4a_0a_2}}{2a_0}$$

三次和四次方程的根也可用根式表示,但这些公式相当复杂,不便于计算。五次以上的代数方程的根已不可能用根式表示,因此对于三次以上的代数方程求根,一般都用数值算法。由于高次方程是非线性方程,因此,求解非线性代数方程的各种数值解法,如牛顿法及其他迭代法均是高次方程的有效解法(参见**非线性代数方程组数值解法**)。但也还有针对高次方程特点的一些特殊解法,较重要的有劈因子法、伯努利法、卢斯表格法和圆盘迭代法。

劈因子法用来求多项式 $P_n(x)$ 的一个二次因子,然后通过解二次方程求得高次方程(1)的一对复根或两个实根。该方法首先用一个近似的二次因子 $w(x) = x^2 + ux + v$ 除 $P_n(x)$,得商 $q(x)$ 及余式 $r(x) = r_1x + r_2$, r_1 及 r_2 由 $w(x)$ 的系数 u, v 确定,于是有

$$P_n(x) = w(x)q(x) + r(x)$$

若 $r_1 = r_2 = 0$,则 $w(x)$ 就是 $P_n(x)$ 的一个二次因子。否则,可令 $w^*(x) = x^2 + (u + \Delta u)x + (v + \Delta v)$,用 $w^*(x)$ 除 $P_n(x)$,得余式 $r^*(x) = (r_1 + \Delta r_1)x + (r_2 + \Delta r_2)$,要求 $r^*(x) = 0$,即 Δr_1, Δr_2 满足方程 $r_1 + \Delta r_1 = 0$, $r_2 + \Delta r_2 = 0$,由此可得到修正量 Δu 及 Δv 的联立方程

$$\begin{cases} \dfrac{\partial r_1}{\partial u}\Delta u + \dfrac{\partial r_1}{\partial v}\Delta v + r_1 = 0 \\ \dfrac{\partial r_2}{\partial u}\Delta u + \dfrac{\partial r_2}{\partial v}\Delta v + r_2 = 0 \end{cases}$$

利用已知关系可求出方程系数 $\frac{\partial r_1}{\partial u}$, $\frac{\partial r_1}{\partial v}$, $\frac{\partial r_2}{\partial u}$, $\frac{\partial r_2}{\partial v}$,从而可求得 Δu 及 Δv,得到 $w^*(x)$,它是比 $w(x)$ 更好的二次因子。重复以上过程直到求出收敛的二次因子为止,最后再求二次因子 $w^*(x)$ 的一对根。此法不用复数运算就可求得复根,这是该方法的优点。

伯努利法可用来求方程模最大和最小的根。卢

斯表格法可用于对方程复根的隔离和判断方程在某区域根的个数，并可求出最大实部根。圆盘迭代法在根隔离后可求出全部根，并具有收敛速度快和易于并行等优点。

在高次代数方程求根过程中，往往会遇到病态多项式，其系数的微小变化会引起零点的很大变化。因此，在编程求根时，通常应采用双精度计算。另外，求 $P_n(x)$ 全部零点时，如用降阶办法，求根次序应按根模由小到大次序进行，才不会影响后面求根的精度。

参考文献

1. 清华大学，北京大学《计算方法》编写组. 计算方法(上册). 北京：科学出版社，1974

2. 林成森. 数值计算方法. 北京：科学出版社，1998

（李庆扬　安恒斌）

gaoji yuyan

高级语言（high level language）　不反映特定计算机体系结构的**程序设计语言**。它的表示方法要比**低级语言**更接近于待解问题的表示方法。其特点是在一定程度上与具体机器无关，易学、易用、易维护。当高级语言程序翻译成等价的低级语言程序时，一般说来，一个高级语言语句要对应多条机器指令，相应编译程序所产生的目标程序往往功效较低。

1952 年瑞士数学家 H. Rutishauser 首先提出高级语言的概念，第一个实用高级语言是美国 IBM 公司 J. Backus 等人于 1956 年研制的 FORTRAN，1960 年相继公布了一种用于事务处理的语言 COBOL 和算法语言 ALGOL 60。这三种语言经过多次修改，出现多个新的标准文本，至今仍在一些领域内占有一定的地位。随后，有人试图研制一类兼收并蓄多种程序设计语言功能的大型通用语言。例如，IBM 公司设计了汇集型语言 PL/1，希望它取代 FORTRAN 和 COBOL，因此只需支持一个语言。又如，可扩充语言 ALGOL 68 具有很强的上下文有关描述能力。但因这类语言过于复杂，系统规模大，效率低，未能流行。结构化程序设计出现以来，人们已设计出多种符合结构化程序设计的语言，如 PASCAL，Ada 等。

随着计算机应用领域的扩大，产生了函数式、逻辑、面向对象程序设计语言，例如：FP，PROLOG，Smalltalk 等等（参见**程序设计语言**）。

高级语言种类千差万别，但是，一般说来基本成分有四种。①数据成分：用以描述程序中所涉及的数据；②运算成分：用以描述程序中所包含的运算，例如：表达式；③控制成分：用以表达程序中的控制构造，例如：条件语句；④传输成分：用以表达程序中数据的传输，例如：输入输出语句。

高级语言正向模块化、简明性、形式化、并行化和可视化等方向发展。

参考文献

1. Horowitz E. Fundamentals of programming language. Rockville: Computer Science Press, 1983

2. Sammet J E. Programming language: history and fundamentals. Englewood Cliffs, NJ: Prentice Hall, 1969

（郑国梁　徐宝文）

gaojie luoji

高阶逻辑（high-order logic）　将**一阶逻辑**的阶数推广至高阶的逻辑。一阶逻辑中的谓词、函词称为一阶谓词、一阶函词，其变元符号都是个体变元符号，量词仅作用于个体变元，这限制了一阶逻辑的表达能力。去掉对变元、谓词、函词的上述限制，就发展成为高阶逻辑。高阶逻辑与类型论有紧密联系。

可以从多方面来推广一阶逻辑。如允许量词不仅可用来约束个体变元，而且也可用来约束谓词变元及函词变元，即允许量词的指导变元不仅可以是个体变元，也可以是谓词变元及函词变元。这样就得到了通常所说的二阶逻辑。例如，$\exists A \exists B(A(x) \to B(x))$ 便是一个二阶逻辑公式。

如果对含有自由谓词变元、函词变元的公式利用概括原则抽象出一个新谓词，那么这个新谓词就是一个以谓词、函词为变元的二阶谓词。例如，对于含二元谓词 R 的公式

$$\forall x \forall y(R(x,y) \to R(y,x))$$

根据概括原则可抽象出一个表示对称性的谓词 sym，即

$$\mathrm{sym}(R) \leftrightarrow \forall x \forall y(R(x,y) \to R(y,x))$$

一般地，由含有至高为 n 阶的自由谓词变元、函词变元的公式根据概括原则抽象出来的谓词、函词便是 $n+1$ 阶谓词、函词。这样可以逐步得到更高阶的谓词、函词，再添入以各阶谓词变元、函词变元为指导变元的量词，更高阶的逻辑便也逐步得到了。但要注意，二阶逻辑中只是含有一阶谓词、函词（自由的及约束的），含二阶谓词、函词的便是更高阶的逻辑了。依照 A. Church 的划分，如果在系统中最高阶的谓词及函词（设为 n 阶）均是自由的，便叫做

$2n-1$ 阶逻辑,最高阶(设为 n 阶)的谓词及函词有约束的,便叫做 $2n$ 阶逻辑。

高阶逻辑有比一阶逻辑更强的表达能力。例如,一阶逻辑不能表达有穷集合的概念,在高阶逻辑中却可用如下公式表达:

$$\forall f(\forall x \forall y(f(x)=f(y) \rightarrow x=y) \rightarrow \forall x \exists y\, f(y)=x)$$

因为一个集合是有穷集合当且仅当该集合上的单射必是满射,所以该公式解释为真命题当且仅当论域是有穷集。

高阶逻辑失去了一阶逻辑的某些良好性质。逻辑有效式的集合不是递归可枚举集,紧致性定理也不再成立,因此不存在可靠且完全的公理系统。基数定理也不成立。

集合论可以表述全部数学,但往往表述得十分复杂难懂,用高阶逻辑表达要简单明了得多。例如,在集合论中用十分冗长的公式才能表述有穷集合的概念。在集合论中对于各阶函数、关系等不加区别,组成一个共通的域,因此对概括原则的使用必须施加适当限制,否则极易产生悖论。而在高阶逻辑中,可以无条件地使用概括原则而不致引起悖论。高阶逻辑对于证明论、递归论(特别是计算复杂性)、公理集合论和一阶逻辑中的证明的研究都是很有意义的。

参考文献

Monk J D. Mathematical logic. New York:Springer-Verlag,1976

(何自强　王戟)

gaosu huanchong cunchuqi

高速缓冲存储器(cache)　位于**中央处理器**与**主存储器**之间,对程序员透明的一种高速小容量存储器。简称高速缓存。

高速缓冲存储器是存储器层次结构的最顶层,通常采用半导体静态存储器作为存储介质,以缩短存取周期。静态存储器虽然速度快,但成本高,占用的器件面积大,不适合大量配备。为了满足应用的需求,通常采用成本较低、集成度更高的半导体动态存储器介质构建容量相对较大但访问时间相对较长的主存储器作为高速缓存的下层。在配备有高速缓存的计算机中,每次访问存储器都先访问高速缓存,若欲访问的数据在高速缓存中,则访问到此为止;否则,再访问主存储器,并把有关数据取入高速缓存。这样,如果大部分访存都能在高速缓存中得到满足,则这个由高速缓存和主存储器构成的一体化存储系统不仅具备了主存的容量,而且可以提供接近高速缓存的存取速度,兼顾了性能和成本的双重需求。高速缓存的设置是所有现代计算机系统发挥高性能的重要因素之一。

高速缓存的工作机制体现了局部性原则。所谓局部性,是指程序访问代码和数据的不均匀性,它包括:①时间局部性:如果某位置已被访问,则该位置很可能在短时间内还要再被访问;②空间局部性:如果某位置已被访问,则其邻近位置很可能还要被访问。因此,只要程序有较好的访存局部性,高速缓存就能发挥作用。

命中率和平均访存时间

高速缓存的组织及对高速缓存的访问都是以块(也称为行)为单位的,通常 1 块包括 1 个或多个字。块也是高速缓存与主存储器交换数据的最小单位。访存时,若能在高速缓存中找到所需数据,称为高速缓存命中,否则称为缺失。命中次数与访存总次数的比率称为命中率,而缺失次数与访存总次数的比率称为缺失率。显然,缺失率 = 1 - 命中率。命中时的访问时间(包括确定是否命中的时间)称为命中时间,缺失时,将主存块替换入高速缓存并提供给中央处理器所需的时间称为缺失损失。缺失时的访存时间等于命中时间加上缺失损失。

命中率是评价高速缓存性能的一个重要指标,它不仅依赖于高速缓存的组织结构,而且与应用程序的局部性状况有关。由于命中率是与硬件速度无关的参数,所以它不能单独用于高速缓存的性能评估。更准确的指标是平均访存时间(命中时间 + 缺失损失 × 缺失率)。例如,根据空间局部性原则,把块变大有助于提高命中率,但同时也增加了缺失时的数据传送时间,所以块的大小应有利于缩小平均访存时间,而不仅仅是为了提高命中率。平均访存时间的单位可以是绝对的(如 2.5ns),也可以是相对的(如 2 个时钟周期)。

块映射策略

当需要将来自内存的新数据块装入高速缓存时,由块映射策略决定它的存放位置。根据内存块在高速缓存中可能的存放位置,块映射策略可分成如下 3 类:

(1) 直接映射　每个块在高速缓存中只能有 1 个位置。该位置通常由求模运算得到,即:

高速缓存内块号 = 块地址(mod 高速缓存中块数)

(2) 全相联映射　块可以放在高速缓存中的任意位置。

(3) 组相联映射　高速缓存分为若干个组,块先映射至组,在组中的位置任意。块向组的映射通常也采用取模运算。块地址中对应于组号的若干位称为索引。若组中有 n 块,则称高速缓存是 n 路组相联的。

高速缓存对每个块都设立 1 个标志域,其中包括块地址和块内数据是否有效的信息。为了实现快速访问,在相联映射型高速缓存中,总是同时查找所有的标志(亦称为相联存储器或按内容寻址存储器)。

块替换策略

块替换策略是高速缓存设计的另一个重要方面。在缺失时,必须将欲访问的主存块取入高速缓存,相应地,新取入的块将替代高速缓存中已有的某一块。对于直接映射高速缓存,只有 1 个块可供替换,所以也无须特别的策略。对于相联映射型的高速缓存,若高速缓存或组内有无效块,则可直接替换;若不存在无效块,则需要采用一定的策略,以避免被替换的块在短时间内又被取回高速缓存,即造成所谓的颠簸现象。常用的块替换策略有随机和最近最少使用(LRU)两种。LRU 策略利用了时间局部性,可能实现较高的命中率,但当需要追踪的块较多时,实现起来比较复杂。

写策略

高速缓存的写操作可以采用如下两种策略:①直写:数据不仅写入高速缓存,同时写入下层的主存储器;②后写:数据只写入高速缓存,只有当相应块被替换时才写回主存。对后写型高速缓存,标记中应增加一个写过位,以区分被写过的块与未被写过的块。这样,发生替换时,只需将被写过的块传送回主存。后写方式有利于提高写的速度,并减少访问内存的次数,但会增加缺失时间,因为缺失时必须等被替换的写过块写回主存后,才能从主存将欲访问的块取入高速缓存。直写较易实现,且主存中数据总是与高速缓存中数据一致。为了提高写速度,减少中央处理器的等待时间,直写高速缓存常采用写缓冲技术,即中央处理器在数据写入高速缓存和写缓冲器后,就认为写操作已经完成,可以执行后续指令,另由单独配备的控制逻辑将写缓冲区的内容写入主存。

当写数据缺失时,亦可采取两种策略:①取写:先把块取入高速缓存,再重复命中的写过程。后写型高速缓存常用此方案。②绕写:只修改主存内容而不取入高速缓存。直写型高速缓存常用此方案。

多级高速缓存

当高速缓存缺失时,直接访问主存会增加程序的执行时间。在单处理机系统中,高速缓存的缺失主要源于下列 3 种情形:

(1) 首次装入　首次访问某块时,必然导致缺失。增加块内字节数有利于减少此类缺失。

(2) 容量不足　高速缓存中不能放下程序运行所需的所有内存块。只有增加高速缓存容量才能解决。

(3) 组内冲突　映射到同一组的块数超过组内可容纳的块时,发生组内冲突。显然全相联高速缓存中没有这类缺失,但块数较多时,全相联是最昂贵且访问速度最慢的。

为了争取速度,高速缓存总是用最快速的静态随机存储器(SRAM)实现,所以增大容量的成本是很高的。而且,增大容量后,访问时间会相应拉长。因为离中央处理器最近的高速缓存的访问延迟直接决定了处理器的工作频率,所以不可能将这个缓存做得很大。为了实现高速缓存与主存储器在容量和速度间的平滑过渡,可以采用多级高速缓存的方案,即在前述与处理器直接耦合的高速缓存和主存间增加 1 级或多级速度稍慢但容量较大的高速缓存。因为第二级高速缓存的作用主要是用来减少第一级高速缓存的缺失损失,并消除因容量不足导致的缺失,它的设计可以采用不同于第一级高速缓存的策略,如采用较高的相联度以提高命中率,使用较长的块来开发空间局部性等。早期的二级缓存都放在处理器芯片之外,近年来,二级缓存已集成到处理器芯片内,甚至有些处理器芯片上还集成了三级缓存。

另外,考虑到程序访问指令和数据的不同特点,可以设置专门存放指令和专门存放数据的高速缓存,且它们可采用不同的容量和策略。

虚地址高速缓存

通常,中央处理器给出的访存地址是逻辑地址(或虚地址),它必须由地址转换机构转换成物理地址后才能访问主存。为了减少访问高速缓存的命中时间,许多系统采用了虚地址高速缓存。这类高速缓存中的地址标志是虚地址,中央处理器给出的地址不经转换就可用于高速缓存访问,而同时该地址可由地址转换机构转换成物理地址,从而一旦高速缓存缺失,可立即用物理地址访问主存。虚地址高速缓存带来的一个问题是,当进程切换时必须清除高速缓存,因为不同进程的虚地址空间是相同的。这对系统(尤其是需要对上下文频繁切换的系统)

性能有一定的影响。把进程号加入标志并参与标志比较，可以避免在进程切换时清空缓存。

作为慢速存储器的自动高速代理，高速缓存的思想也应用在计算机系统的其他方面。下面是一些典型的例子：

（1）转换检测缓冲器（快表）　在虚实地址转换过程中，为了快速访问内存中的页表项而设置的一个专用高速缓存（参见**转换检测缓冲器**）；

（2）文件高速缓存（file cache）　操作系统中的文件系统为了缩短访问磁盘文件块的延迟，在内存中开设的缓冲区；

（3）磁盘高速缓存（disk cache）　为了缩短访问磁盘数据块的延迟而在磁盘控制器中设置的缓冲存储器；

（4）网页高速缓存（web cache）　为了缩短访问网页的延迟，浏览器或代理服务器在本地磁盘中开设的缓冲区。

参考文献

1. Jim Handy. The Cache Memory Book. San Diego: Academic Press Inc., 1993

2. Hennessy J L, Patterson D A. Computer architecture: a quantitative approach. 5th ed. Morgan Kaufmann, 2011

（唐志敏）

gaosu huanchong cunchuqi yizhixing

高速缓冲存储器一致性（cache coherence）

在采用层次结构存储器（参见**分层存储器体系结构**）的计算机系统中，保证**高速缓冲存储器**中数据与**主存储器**中数据相同的机制。

在单处理机系统中，高速缓存数据与主存数据的不一致主要是由输入输出（I/O）操作（如直接存储器存取）引起的。在几乎所有的计算机系统中，I/O 数据总是直接从主存进出而不经过高速缓冲存储器，从而，①在使用后写型高速缓冲存储器的系统中，输出数据时，I/O 系统看到的可能是主存中的旧数据（新数据在高速缓冲存储器中）；② 输入数据时，**中央处理器**只能看到高速缓存中的旧数据（新数据在主存中）。为了避免这样的数据不一致，需要硬件或**操作系统**的干预。例如，在进行输出操作时，由操作系统将有关地址的数据移出高速缓存，或者由硬件检查高速缓存中的相应标志，发现与输出数据地址相同时，立即写回主存；在进行输入操作时，可由操作系统保证输入数据区不被缓冲，或先将高速缓存中的有关内容清除。

在使用虚地址高速缓存的系统中，系统软件将两个不同的虚地址映射到同一物理地址的别名现象会导致高速缓存数据与主存数据的不一致以及高速缓存中出现同一数据的两个副本。通过对别名进行简单的限制就可避免这种形式的不一致。

共享存储多处理机的高速缓冲存储器一致性

高速缓存一致性问题主要出现在共享存储器的多处理机系统中。多处理机对高速缓存一致性的要求可表述为：任意一次取数操作得到的结果都必须是最近一次对该数据进行写操作时写入的值。由于互连网络的延迟和系统中各种缓冲和优化机制的影响，“最近”的含义实际上是无法准确认定的。共享存储器的一致性模型研究的就是从程序员的角度如何容忍对“最近”一词的模糊认识；而高速缓存一致性协议则在此基础上讨论如何正确维持同一数据的多个相同副本。

虽然共享数据的多副本现象是高速缓存数据不一致的主要原因，但有时非共享数据也会出现不一致的问题。其原因除了前述的 I/O 操作外，还有进程迁移。若一个进程在处理机 A 上访问了变量 X，迁移到处理机 B 后又修改了 X，则当它再回到 A 时，就有可能读到 A 的高速缓存中存在着的 X 的旧值。如果强制每次上下文切换时都清除高速缓存的内容，就不会出现这种情况，但系统性能将会下降。更好的方法是在高速缓存数据的标志中增加进程标识符（PID）信息。

因多副本导致的高速缓存不一致可用下例说明（考虑 3 个处理机的情形）：

（1）处理机 A 和处理机 B 分别读变量 X（设其初值为 12），则 X 分别进入 A 和 B 的高速缓冲存储器；

（2）处理机 A 将 X 修改为 16，则 cache A 中 $X=16$，cache B 中 $X=12$；

（3）处理机 A、B、C 分别读 X，则 A 得到 16，B 得到 12，C 得到的值取决于高速缓存的写策略：若采用直写策略，C 得到存储器中的新值 16，若采用后写策略，则得到存储器中的旧值 12。

避免出现这一问题的最简单方法是不允许共享数据进入高速缓存，一些早期的系统（如伊利诺依大学研制的 CEDAR）就采用了这一方式。但这对系统性能的影响很大，尤其是在分布式共享存储器系统中，不缓存共享数据就无法利用远程访问中的局部性，从而大大增加了访存的延迟时间。

一致性协议

彻底解决上述问题的途径是采用硬件或软件的机制(协议),保证当某一个处理器更新高速缓存中的共享数据时,主存储器和持有该数据副本的其他高速缓存能及时知道这一动作。下述通用协议就能实现这一目标。

(1) 读共享数据时,若缺失,则先询问所有其他高速缓存,以得到最近被写入的数据。

(2) 写共享数据时,不论是否命中,都要求每个高速缓存检查自己是否持有该数据的副本。对于其他高速缓存中的副本,可采用如下处理方式之一:①写无效:将所有副本置为无效状态;②写广播或写更新:将写入的新值送给所有持副本的高速缓存。

虽然通用协议能保证高速缓存数据的一致性,但其广播特性对系统性能有较大的影响。事实上,一致性维护很容易使互连网络带宽达到饱和。

实际使用的高速缓存一致性协议可分为3种类型,即监听协议、目录机制和编译制导的一致性维护。下面分别简单介绍这3种协议。

(1) 监听协议　主要用于总线结构的多处理机。每个处理机都配备独立的监听硬件,监视出现在总线上的存储器访问或一致性维护操作,并检测自己高速缓存内的数据是否被其他处理机修改。如果发现高速缓存中数据已被修改,则或者无效或者更新本地数据。在写无效时,下次访问该数据时高速缓存将缺失,从而处理机可从主存或其他高速缓存中获得正确的数据。监听机制利用了总线的广播特性,实现较简单,但由于总线本身的限制,一般只适用于小规模的多处理机系统。也有人考虑将这一协议用于层状总线系统和环状系统。在监听协议中,与一致性有关的控制信息存放在高速缓存中,从而也被称为是基于高速缓存的一致性协议,以区别于基于目录的协议。

多级互连网络、二维或三维网格、超立方体和胖树等连接网络具有比总线高得多的总带宽,但访问存储器的延迟时间也加长了,因而对本地高速缓存的性能也有更高的要求。由于在这些互连网络中处理机已不再具有监视所有存储器操作的能力,监听协议无法适用。目录机制及编译制导机制就是为这类系统设计的。

(2) 基于目录的高速缓存一致性协议　在这种协议中,增加了一个称为目录的数据结构和相应的管理机制。每个内存块都对应一个目录项,用于记录该内存块的状态(是否被缓存、是否被修改、哪些处理机的高速缓存中存有数据副本等)。这样,处理机通过目录就可以得知它的访存动作是否会导致数据的不一致。处理机写数据前,必须从目录处获得对某一数据块的独占访问权。而目录在接到处理机对独占访问权的申请后,先向所有持有该数据块副本的处理机发无效命令,直到收到所有这些处理机的应答信号后才将独占访问权交给提出申请的处理机。当处理机试图读被别的处理机独占的数据块时,它先向目录请求读缺失服务。目录于是向持有独占副本的处理机发命令,要求它将数据写回存储器(同时收回独占访问权)。当收到该数据后,再将它转发给请求读数的处理机。

目录机制的具体实现可能在上述过程的基础上略有变化。例如,系统可以选择写更新而不是写无效的策略。不同于监听机制(其中状态总位数正比于所有高速缓存容量之和),目录机制中需维持的信息正比于系统中的存储器总容量,且每个存储器数据块除状态位外,还必须配备记录该块数据在哪些处理机的高速缓存中有副本的存储位。因此,目录信息的存储位置及存储策略(集中还是分布)是一个需要仔细斟酌的问题,一般可采取全映象或位向量、有限指针、链式目录等方式。另外,在允许弱一致性模型(参见**共享存储**)的系统中,对一般共享数据(有别于同步变量),可以不等收到所有的无效应答信号就开始写操作,以提高系统的整体性能。

(3) 编译制导的一致性协议　在编译时确定哪些高速缓存数据块会过时,并生成特殊的指令插入代码段中,以防止处理机使用这些可能过时的数据(通常采用将有关数据清除出高速缓存的办法)。与目录机制相比,简单的编译制导策略有可能导致过多的高速缓存数据无效,从而降低存储系统的性能。

性能与成本

多处理机中选择高速缓存一致性机制的主要考虑是性能和开销方面的权衡,一方面,允许远程数据进入高速缓存可以有效地缩短访存延迟,另一方面,一致性协议本身会带来一定的系统开销,如处理机长时间地等待远程访问的结束,一致性消息的传送占用了互连网络的带宽等。另一个需要考虑的因素是设计和实现的成本,如目录信息的存储量和控制逻辑的复杂度、一致性协议实现正确性的验证等。此外,高速缓冲存储器数据块的大小对一致性机制的性能也有较大的影响。原则上,大的数据块可以

充分地利用数据的空间局部性，发挥高速缓冲存储器的预取功能，但也带来了较难处理的假共享问题，即当两个处理机分别频繁地写同一块中的两个字时，一致性机制认为它们共享同一数据块，导致处理机间来回地发无效命令并进行数据传送，不仅影响了性能，而且占用了宝贵的网络带宽。采用支持弱一致性的编程模型，把同步操作和一致性维护操作相结合，可有效地降低一致性维护带来的开销。

参考文献

1. Hwang K. Advanced computer architecture: parallelism, scalability, programmability. New York: McGraw-Hill Inc., 1993

2. Hennessy J L, Patterson D A. Computer architecture: a quantitative approach. 5th ed. Morgan Kaufmann, 2011

（唐志敏）

gaosu shuzi xinhao chuanshu

高速数字信号传输（high speed digital signal transmission） 在现代计算机和数字设备中由于数字电路速度的提高，信号在互连线上的传输时间可与电路本身的级延迟时间相比拟时所产生的信号传输问题。它主要是研究和解决高速数字信号在传输时发生的反射和串扰两大问题。由于传输线自身特性（特性阻抗、传输速度、损耗等）的不同，传输线沿途分布的负载的不同，多段传输线接续的不均匀和端接不匹配等复杂因素所引起的反射，使数字信号在传输中产生畸变，表现为边沿成台阶状，或产生上冲和下冲。信号在传输线中传输，通过互感和电容对相邻传输线产生串扰，也使被串扰的信号发生畸变。反射和串扰使信号发生的畸变会增加电路的级延迟，影响系统速度，甚至会使电路产生误动作，影响系统的可靠性。所以研究和掌握反射和串扰的规律，制定相应的工程规范对其进行控制，对于保证系统的速度和稳定可靠是至关重要的。

高速数字信号传输时的反射

（1）传输线的特性阻抗　在数字电路的应用中往往把传输线看成理想的无损传输线，它的特性阻抗

$$Z_0 = \sqrt{\frac{L_0}{C_0}} \qquad (1)$$

其中，L_0为传输线单位长度的电感，C_0为传输线单位长度的电容。可以认为L_0和C_0与频率无关，特性阻抗呈现为与频率无关的纯电阻。

电信号在传输线中以接近于光速的速度传输，经过一定长度的传输线有一定的延迟。经过单位长度传输线的延迟t_{pd}是单位长度传输线的电容和电感的乘积的平方根，它由传输线介质的相对介电常数ε_r决定，即有

$$t_{pd} = \sqrt{L_0 C_0} = 0.333 \times 10^{-8}\sqrt{\varepsilon_r}\,\text{s/m} \qquad (2)$$

常用的传输线有同轴电缆、带状电缆、印制线和双绞线等，它们所用的介质材料决定了相应的传输延迟。同轴电缆的t_{pd}为4～5 ns/m，Z_0为50 Ω、75 Ω和125 Ω。双绞线的t_{pd}为4.5～6 ns/m，Z_0为50～100 Ω。常用的印制电路板板材的ε_r在3.0～6.2之间，t_{pd}为5.78～8.3 ns/m。

传输线的特性阻抗Z_0是一个重要的系统参数。Z_0低的系统稳定性较好，但电路的驱动功耗大。通常不同的逻辑电路选择不同的特性阻抗。ECL电路常选用Z_0＝50 Ω，晶体管逻辑（TTL）电路常选用Z_0＝75 Ω。一旦确定了Z_0，在制造和选择传输线时，要将Z_0控制在一定范围内。

（2）传输线的多次反射　设用内阻为R_0的信号源$E(t)$驱动特性阻抗为Z_0的传输线，传输线终端接有阻抗为Z_L的负载。传输线始端S的入射电压

$$U_{S0}(t) = \frac{Z_0}{Z_0 + R_0}E(t) \qquad (3)$$

$U_{S0}(t)$经长度为l的传输线向终端前进，经过$T_d = lt_{pd}$的延迟到达终点L。当负载阻抗Z_L等于特性阻抗Z_0时，入射电流全流入Z_L，终点L的端电压等于入射电压$U_{S0}(t)$，没有反射产生。这是波形完全不畸变传到终点的理想情况。当负载阻抗不等于特性阻抗时，就产生第一次反射，终端电压等于入射电压和反射电压之和。第一次反射电压U_{LF1}沿相反方向又经T_d入射到始端S，由于始端的内阻一般不等于特性阻抗，又产生新的反射电压U_{SF1}，这时始端电压为$U_{S1} = U_{LF1} + U_{SF1}$。始端的第一次反射电压$U_{SF1}$还继续传向终端，再产生第二次反射。这个过程一直继续下去，直到第n次反射电压接近零，波形达到稳定为止。始端电压是$U_{S0}(t)$和多次反射后形成的始端电压在时间轴上的叠加，即始端所有入射电压和反射电压的总和。同样，终端电压是终端的多次入射电压和反射电压对时间的叠加。

传输线沿线各点的数字信号是驱动信号和多次反射叠加形成的，反射程度决定了信号畸变的形状和大小。传输线的特性阻抗、传输速度和长度、多段传输线的接续方式和均匀性都直接影响到反射。

（3）传输线的匹配终端　数字电路既是驱动电

路又是负载电路，它的输出阻抗构成传输线驱动电路的内阻 R_0，输入阻抗构成传输线的负载 Z_L。数字电路（包括 TTL，射极耦合逻辑 ECL 和互补金属-氧化物-半导体 CMOS 电路）的输入阻抗和输出阻抗都是非线性的。输入阻抗的电阻成分在 0 态和 1 态都呈几十 kΩ 的大电阻，在开关过渡区则在百 Ω 数量级。电抗成分为电容，约几个 pF。ECL 电路的输出阻抗与 TTL 电路、CMOS 电路不同，ECL 电路采用射极跟随器输出，在高电平（1 态）和低电平（0 态）时的输出阻抗比较接近（均为数 Ω），TTL 电路和 CMOS 电路的不同电平时的输出阻抗则有较大差别。

为了吸收反射，减少传输线不匹配和沿线负载的不良影响，普遍采用匹配终端的方法。常用的匹配终端的方法有以下 5 种。①串联电阻　适用于负载集中在线的终端的情况。电阻串接在驱动源附近，其阻值为负载传输线特性阻抗和驱动源内阻之差。②并联电阻　此方法应用广泛。将阻值等于负载传输线特性阻抗的电阻一端接在传输线终点上，电阻另一端接地或接电源 V_T。在 ECL 电路中，$V_T = -2$ V；在 TTL 电路中，$V_T = +3$ V 或 +5 V。③分压电阻　适用于 TTL 电路。传输线终点上接有两个电阻，其中一个电阻另一端接 +5 V，另一个电阻的另一端接地。此方法本质上等效于并联电阻，常用于时钟信号线和总线上。④阻容网络　在 TTL 电路和 CMOS 电路中能很好地工作。此方法是将电阻和电容串联接在传输线终点与地之间，电容值在 200～600 pF 范围内。电容与电阻形成的时间常数（RC 值）必须大于负载传输线延迟的两倍。⑤二极管网络　常用于差分网络中，它能将过冲限幅在 1 V 以内，功耗小。5 种方法各有利弊，要结合具体条件通过实验合理选择。

为了减少反射，在布线设计中推荐沿线分布负载，避免 T 形分叉线，管座和过孔应分别增加 5 pF 和 2 pF 负载；要计入分布电容的计算；建议少用管座，减少过孔；在高性能的数字系统中要控制特性阻抗。减少反射的根本出路在于缩短传输线长度，提高集成电路的集成度，采用高密度组装技术。

时域反射仪（TDR）是观察和测试反射的仪器，它能显示传输线不连续的地点和程度。

高速数字信号传输时的串扰

高速信号在传输线上传输时，由于互感和电容的存在，在邻近传输线中感应的噪声信号。设有一个数字脉冲在两根相邻的特性阻抗为 Z_0 的平行传输线之一中传输，两线互相耦合的电容和互感为 C_m 和 L_m，在驱动线上脉冲到达某一点 a 引起电容电流 i_C 在感应线中流动。它分成数值相等方向相反的两部分，且各向着感应线的两端流动。驱动脉冲前进的方向称为前向，其相反方向为后向。根据楞次定律，感应线上还感应出电感电流 i_L 阻止驱动电流的流动，i_L 总是流向后向。因此在 a 点有 $i_L + i_C$ 流向后向，$i_L - i_C$ 流向前向，i_L 和 i_C 都正比于驱动电流函数的导数。

在感应线的后向终点的端电压与驱动源极性相同，电流 $i_L + i_C$ 在负载上形成了后向串扰电压。在感应线的前向终点，由于 i_L 和 i_C 的方向相反，但大小不会相等，电流 $i_L - i_C$ 也会产生一个前向串扰电压。长线的后向串扰电压的幅度正比于驱动电压的幅度，宽度等于传输延迟的两倍，即 $2T_d$。前向串扰电压幅度正比于驱动信号对时间的导数和耦合长度，宽度等于驱动信号的上升时间。

在工程中，传输线两端的匹配终端情况和沿线负载不同，再加上传输线是非均匀介质，串扰会更加复杂。

在实际应用中减少串扰的方法是限制耦合长度和增加平行线间距离。在多层印制板中每两层信号线层间用一层地和电源层隔离，两层信号线互相垂直。连接器的引线之间的串扰不可忽视，因而在高速应用中要用电缆中一部分引线两端接地，以便隔离引线间的串扰。带状电缆的信号线中要有一定比例的地线安插在信号线之间，多条带状电缆应避免平行重叠放置。　（陈鸿安）

gaoxiaoneng jisuanji

高效能计算机（high productivity computer）

在高性能计算机基础上发展起来的一种计算机系统。传统的高性能计算机强调系统运行程序的性能，衡量一个系统优劣的典型方法是运行公认的基准测试程序，例如 Linpack（参见**基准程序**）。与高性能计算机不同，高效能计算机强调除了运行性能之外的其他系统指标，包括低系统成本、程序开发的高效率、方便地支持已有程序的移植以及系统的高可靠性等，并将这些指标的综合称之为系统的效能。

高效能计算机的概念来源于美国 DARPA 所支持的高效能计算系统（HPCS）研究计划。HPCS 计划于 2002 年启动，执行周期为 2002—2010 年。该计划的宗旨是，以最经济的手段，开发能够支持美国国家安全和科研与产业发展的新一代计算机系统。

HPCS 从 4 个方面定义了高效能的含义：①以经济手段实现系统高性能；②具有良好的系统可编程性；③具有良好的程序可移植性；④系统的高鲁棒性。

随着高效能计算机规模的不断提高，以更经济的手段实现系统高性能的需求日趋迫切。高效能计算机强调要以低成本实现系统的高性能。这里的成本不仅仅指系统开发的成本，还包括系统运行的成本。低成本、高性能的指标意味着不能靠简单地扩大规模、增加部件数量来实现系统的高性能，而是要通过创新的体系结构、创新的器件与技术、硬件和软件的有机协同来提高系统性能。降低系统运行成本的一个重要方面是降低系统的能耗，随着高性能计算系统规模的扩大，系统的耗电已经成为其运行的主要成本，因此，低能耗成为高效能计算机的重要指标和特征。

良好的系统可编程性是提高程序开发效率的关键因素。高效能计算机不仅强调系统运行的高效率，还强调应用开发的高效率。改善系统的可编程性有助于提高编程效率，缩短程序开发的时间，从而实现高效率开发应用的目标。提高系统可编程性的途径主要是发展更加适应机器结构的新的并行编程模型和并行编程语言，通过模型和语言的表达能力和特有结构，使得编程人员能专注于所要完成的应用逻辑，而无须关心机器的特殊物理细节，从而提高编程的效率。评价系统可编程的一个指标是实现应用解决方案的时间，时间越短，可编程性越好，以这个指标来评价不同的编程模型和编程语言的优劣。

实现良好的程序可移植性是高效能计算机的另一个重要目标。应用系统的复杂性决定了应用软件开发的成本很高，不可能每换一种机型就重新开发一遍应用软件，必须保证现有程序在新系统上能顺利移植并高效执行。不论新的计算机系统采用何种新的体系结构和何种实现技术，对应用而言，仍能保持其发展的连续性和演进的平滑性，这不仅能有效地提高应用开发效率，同时也保护了用户已有的软件投资。

系统的高鲁棒性对于大规模计算机系统至关重要。随着系统规模增大，可靠性已经成为瓶颈。由数以千万个器件构成的大规模计算机系统的平均无故障时间很短，可能只有几个小时。如何提高这类系统对异常和故障的容忍能力，实现高度鲁棒的计算系统，保证用户程序的连续正常执行，是高效能计算机设计与实现必须考虑的重要因素。高效能计算机在器件、部件、系统多个层次采取可靠性设计措施，以硬件和软件相结合的方法，实现系统的长时间不间断运行。

国际高效能计算机的代表性研究计划是美国的 HPCS 计划。该计划 2002 年开始，分 3 期实施。一期主要研究高效能计算机的关键技术，二期对于关键技术进行小规模验证，三期开发高效能计算机系统。IBM 和 Cray 是仅有的两个全程参与 HPCS 计划的公司，IBM 的高效能计算机代号为 PERCS，系统采用 IBM 的 Power 7 处理器、AIX 操作系统、新型编程语言 X10 以及全局并行文件系统等技术实现，以达到高效能的目标。第一个提交的实用 PERCS 系统是于 2011 年在美国国家超级计算中心(NCSA)部署的万万亿次的“蓝水”超级计算机。Cray 的高效能计算机代号是 Cascade，系统将基于 Intel 最新的多核处理器、Cray 公司全新的互连 Aries、面向 Cascade 的新型编程语言 Chapel 等实现。Cascade 将采用异构的体系结构，即除了通用处理器构成的计算部件外，还将采用由图形处理部件 GPU 或 Intel 的新型众核处理器构成的加速部件。Sun 公司尽管没有能参与 HPCS 的第 3 期研究，但是这个以技术创新见长的公司在 HPCS 前两期实施过程中提出了众多创新性的技术，包括以电容变化传输信号的接近互连、能实现电子和光子集成的硅光子技术、对象存储、区间计算以及新的编程语言 Fortress，这些技术对高效能计算机的发展有着深远的影响。

中国的高技术研究计划(863 计划)从 2006 年起开始高效能计算机的研发，在该计划支持下，在异构体系结构、高性能互连、高密度低功耗计算节点等方面取得了进展，研制成功了天河一号、曙光星云、神威蓝光、联想深腾等高效能计算机系统，在高效能计算机关键技术和系统实现方面取得了长足的进步。

参考文献

1. http://www.highproductivity.org/

2. Lusk E, Yelick K. Languages for High-Productivity Computing: the DARPA HPCS Language Project. Parallel Processing Letters, 2007, 17(1): 89-102

(钱德沛)

gaoxingneng jisuan

高性能计算(high performance computing)

用高速计算机系统解决复杂问题的计算方式。由于高性能计算通常使用并行计算技术，因此高性能

计算体现为各种不同形式的并行计算。高性能计算可以解决原来单靠理论方法或者实验方法无法解决的问题,并且在传统的理论与实验方法之间架起了相互联系的桥梁。高性能计算在现代科学研究、国民经济和社会发展中的作用越来越重要。

执行高性能计算的计算机称为高性能计算机。高性能计算机发展的几个阶段和不同的体系结构密切相关。

高性能计算机发展的第一个阶段是 20 世纪 70 年代中期以 Cray 1 超级计算机为代表的向量计算机(参见**向量计算**),这种超级计算机主要由特殊的向量处理器和专门的操作系统构成。向量计算机后来进化为并行向量处理(PVP)高性能计算机。

第二个阶段是在 20 世纪 90 年代的对称式多处理(SMP)(参见**共享存储**)、**高速缓冲存储器一致性**非均匀存储器访问(CC-NUMA)(参见**共享存储**) 以及大规模并行处理(MPP)(参见**大规模并行处理**)等结构。SMP 与 CC-NUMA 都是共享存储系统,它们具有如下特点:①对称性,即任何一个处理器都可以访问本系统的任何内存单元和外部设备;②单一地址空间,即不存在局部与全局地址空间的问题,所有地址和数据都在一个相同的空间中;③存储器通信,和 I/O 通信相比,可以具有较高的通信效率。CC-NUMA 是分布式共享存储系统,它是 SMP 系统的扩展,既可以保持 SMP 结构的优点,又增强了扩展性。MPP 与 SMP 和 CC-NUMA 的最大不同之点在于其物理内存的分布性,同时其处理器个数也可以比较多,MPP 高性能计算机的处理器可以成千上万。

第三个阶段是 20 世纪 90 年代中后期发展起来的集群式高性能计算机系统(参见**集群计算**),它采用成品化的部件,其特点是性能价格比较高,同时研制周期比较短,有很好的扩展性。它一般是由工作站或者高档微机组成计算结点,通过高速**互连网络**连接起来,采用商品化的操作系统或者免费操作系统与软件工具。另有一种高档次的集群高性能计算机系统,它是以 SMP 或者 CC-NUMA 等高性能计算系统为超结点而构成的系统,可以提供非常高的计算能力。

并行编程模型

高性能计算需要高性能的并行编程模型,主要的并行编程模型有共享变量、数据并行和**消息传递**三种,另外还有面向对象、函数式与逻辑编程模型等。下面介绍主要的并行编程模型。

在共享变量模型中,多个并行成分可以对相同的共享变量进行操作,它具有如下特点:①单一地址空间,由于共享变量与数据在相同的地址空间中,因此不需要显式数据分配,各个进程或线程可通过访问公共存储器中的共享变量来进行通信;②多线程,这需要编程者设计不同的并行执行语句,来实现程序的并行执行;③异步,线程之间需要显式的同步语句,来保证各个线程有正确的执行顺序。这种模型的典型代表是 OpenMP 标准,此外还有 Pthreads 等。和消息传递编程模式相比,共享变量程序的移植性相对较差。

在数据并行模型中,不同处理器同时对不同的数据执行相同的操作,它具有如下特点:①全局名字空间,从概念上说,数据并行模型提供给各个不同处理器的是共享内存,它们具有相同的地址空间,因此不需要消息通信语句,不同处理器之间的通信是通过访问共享内存实现的;②单线程,编程者将数据并行程序看作是单个线程的执行,而不是多线程的执行,具有类似串行程序的控制流;③松同步,即数据并行程序的同步是语句级的,而不是指令级的。针对数据并行编程模式,有相应的并行语言,常用的是高性能 FORTRAN(HPF)。它通过引入标注语句,让编程者负责实现对数据的划分,同时引入一些常见和方便操作的并行结构,用来表达各种数据并行执行形式。这种方式的优点就是可以提高编程的级别,提高编程效率,缺点就是对于一些非典型数据并行问题,一般不易表达,而且性能也不高。

消息传递是另外一种重要的并行编程模型,它的特点是:①多进程,从程序员的角度看,消息传递并行程序表达的是多个进程的并行执行;②独立地址空间,即各个进程都有自己独立的编址空间,不同进程之间相互独立;③异步执行,各个进程之间的同步是通过显式的消息传递语句实现的,它们之间完全可以异步执行。消息传递编程模型的一种常用标准是消息传递接口(MPI)。用 MPI 来编写并行程序,可以取得较高的通信效率,但是编程者必须负责完成不同处理器之间的通信操作,这样就增加了编程者的负担,影响了并行程序的开发效率。

高性能计算工具与环境

高性能计算必须有相应的工具和环境的支持。程序开发工具和环境有助于高效开发高性能的应用程序,包括程序编辑器、并行程序编译器、并行程序调试器、并行系统软件和支撑语言等。

(1) 并行编译器　一个并行编译器主要由 3 部

分组成：流分析、程序优化和代码生成。流分析是确定源代码中数据和控制的相关性；优化是将代码变换成与之等效但更好的形式，以利于挖掘硬件潜力，最终达到全局优化的目的；代码生成涉及从一种描述转换到另一种中间形式的描述。主要的并行编译器有两种，即自动并行编译器和人工辅助并行编译器。并行编译有向量并行、函数并行、指令并行等不同的并行层次。还可为并行计算机开发智能编译器，但难度很大，离实际需求相差甚远。

（2）并行调试器　并行程序的调试和分析很困难。调试的目的是为了获得一个正确的并行程序。目前并行程序调试的技术和手段不成熟，其主要原因在于并行化的程序语句执行的次序是不确定的。这种不确定性意味着特殊的机器指令执行序列是不可重现的，因此难以跟踪观察。主要的调试手段有断点调试、事件分析和重放。

（3）并行程序性能分析　并行程序的性能分析工具主要用来分析并行程序的性能，提供性能参数，指导程序的优化和改进系统的设计。性能分析工具一般分为静态和动态两种：前者采用模拟或分析方法获取源程序中的有关性能数据；后者采用测量的方法收集程序运行中的各种性能数据，即时或事后报告给用户。动态分析提供的数据比较准确，但灵活性较差；静态分析能够针对不同的程序和运行环境给出性能预测，但准确性有待提高。

发展趋势

半导体芯片是高性能计算机的核心器件，微电子技术的革新是50年来推动计算机技术发展的最根本的驱动力。然而，传统的以硅为基础的芯片制造技术的发展不是无限的，由于存在磁场效应、热效应、量子效应以及制作上的困难，因此需要开拓新的芯片制造技术。以新技术为基础的未来高性能计算包括分子计算、光计算（参见**光计算机**）、生物计算（参见**生物计算机**）和量子计算（参见**量子计算机**）等。这些技术离实用还有相当距离，但是由于新技术具有诱人的潜力，因此引起人们的广泛注意。

参考文献

1. 黄铠，徐志伟. 可扩展并行计算：技术、结构与编程. 陆鑫达，等译. 北京：机械工业出版社，2000

2. 陈国良. 并行计算——结构·算法·编程. 北京：高等教育出版社，1999（陈渝　郗志辉）

Gedeer peishu

哥德尔配数（Gödel numbering）　使用素数幂的乘积来表示自然数的任意有限序列 $x_0, x_1, \cdots, x_n$ 的所有方案。表示本身称为序列 $x_0, x_1, \cdots, x_n$ 的哥德尔数，通常记为 $\langle x_0, x_1, \cdots, x_n \rangle$。虽然每种方案都有自己的长处与不足，但是，经常使用的是所谓标准方案。在标准方案中，序列 $x_0, x_1, \cdots, x_n$ 用自然数 $P_0^{x_0} P_1^{x_1} \cdots P_n^{x_n}$ 来表示，其中 P_i 表示第 i 个素数。按标准方案，许多不同的序列可能具有相同的哥德尔数。定义二元函数 E，使得

$$E(i,z) = \mu^z x[\operatorname{div}(P_n(i)^{x+1}, z) = 0]$$

其中，二元函数 div 定义为：若 $x>0, y>0$ 且 x 整除 y，则 $\operatorname{div}(x,y)=1$；否则 $\operatorname{div}(x,y)=0$。$P_n(i)$ 是第 i 个素数。若 z 是序列 $x_0, x_1, \cdots, x_n$ 的哥德尔数，则对每个 $0 \leqslant i \leqslant n$ 有 $E(i,z)=x_i$；对每个 $i>n$ 有 $E(i,z)=0$。函数 E 称为提取函数，值 $E(0,z), E(1,z), \cdots E(i,z)$ 称为 z 的分量。定义一元函数 Lh，使得

$$\mathrm{Lh}(z) = \mu^z n\left[\prod_{i=0}^{n} P_n(i)^{E(i,z)} = z\right]$$

则 $\mathrm{Lh}(z)$ 是整除 z 的最大素数的下标。函数 Lh 称为长度函数，$\mathrm{Lh}(z)$ 的值称为 z 的长度，常记为 $|z|$。

为了对不同的序列提供不同的表示，经常使用标准方案的以下两种变形：在第一种变形中，序列 $x_0, x_1, \cdots, x_n$ 用自然数 $P_0^{x_0+1} P_1^{x_1+1} \cdots P_n^{x_n+1}$ 来表示。按这种方案，许多自然数根本不表示任何序列。在第二种变形中，序列 $x_1, x_2, \cdots, x_n$ 用自然数 $P_0^n P_1^{x_1} P_2^{x_2} \cdots P_n^{x_n}$ 来表示。按这种方案，并非每个自然数都表示某个序列。除此之外，还有许多其他的变形。

哥德尔配数为长度不等的所有有限序列规定了一个顺序。许多重要定理的证明都用到哥德尔配数。

参考文献

Hennie F. Introduction to computability. Boston, MA：Addison-wesley，1977（殷建平）

Gedeer wanquanxing dingli

哥德尔完全性定理（Gödel's completeness theorem）　K. Gödel 于 1930 年证明的谓词演算（参见**一阶逻辑**）具有完全性的定理。谓词演算具有可靠性，即它的每个定理都是逻辑有效式。哥德尔完全性定理说：谓词演算具有完全性，即每个逻辑有效式都是谓词演算的定理。

谓词演算是没有非逻辑公理的一阶理论，可以将哥德尔完全性定理推广到一般的一阶理论。若公式 A 在一阶理论 T 的每个模型中有效，则称 A 在 T 中

有效,记为 $T \models A$。A 是 T 的定理记为 $T \vdash A$。可以证明,对于一阶理论 T 的公式 A,$T \models A$ 当且仅当 $T \vdash A$。人们也把它称为现代哥德尔完全性定理。这个定理成立的根本原因在于:逻辑有效式的集合是递归可枚举集和一阶逻辑具有紧致性(参见**模型论**)。

如果有一个公式 A 使得 A 和 $\neg A$ 都是一阶理论 T 的定理,则称 T 是不协调的,否则称 T 是协调的。哥德尔完全性定理的另一等价形式是:一阶理论 T 协调的充分必要条件是 T 有模型。哥德尔完全性定理说明,一阶逻辑的形式系统完全正确地反映了直观的逻辑推理。它把一阶逻辑的语法性质(公式的可推出性)和语义性质(公式的有效性)联系起来,表明二者是一致的。这为用语义方法研究一阶理论的语法性质,以及用语法方法研究一阶理论的语义性质提供了理论根据,为模型论的诞生准备了条件。由哥德尔完全性定理可以直接导出一阶逻辑的紧致性。由谓词演算的定理集是递归可枚举集和哥德尔完全性定理可以得出,逻辑有效式的集合是递归可枚举集。判断一公式的逻辑有效性是人工智能经常需要面对的问题,判断公式逻辑有效性算法理论基础的埃尔布朗定理,正是一阶逻辑紧致性定理的直接推论。

参考文献

1. 王宪钧. 数理逻辑引论. 北京:北京大学出版社,1982

2. Kleene S C. 元数学导论. 莫绍揆,译. 北京:科学出版社,1984

(何自强)

ge

格(lattice) 一种特殊的偏序集。

设 $\langle L, \leqslant \rangle$ 为一个偏序集。若对任意 $a,b \in L$,L 的子集合 $\{a,b\}$ 都有最小上界和最大下界,则称 L 是一个格,并把 $\{a,b\}$ 的最小上界和最大下界分别记为 $a \cup b$ 和 $a \cap b$,分别称为 a 与 b 的并和 a 与 b 的交。若把 $\cup$ 和 $\cap$ 看作 L 上的两个二元运算,则它们满足以下运算定律:

交换律:对任意 $a,b \in L$,皆有 $a \cup b = b \cup a$ 和 $a \cap b = b \cap a$;

结合律:对任意 $a,b,c \in L$,皆有 $(a \cup b) \cup c = a \cup (b \cup c)$ 和 $(a \cap b) \cap c = a \cap (b \cap c)$;

吸收律:对任意 $a,b \in L$,皆有 $a \cup (a \cap b) = a$ 和 $a \cap (a \cup b) = a$。

反过来,若在集合 L 上定义了满足上述交换律、结合律和吸收律的两个二元运算 $\cup$ 和 $\cap$,则可在 L 上引出一个偏序:对任意 $a,b \in L$,$a \leqslant b$ 当且仅当 $a \cup b = b$(当且仅当 $a \cap b = a$)。可以证明,$\langle L, \leqslant \rangle$ 是一个偏序集,且对任意 $a,b \in L$,$\{a,b\}$ 在 L 中有最小上界和最大下界,即 $\langle L, \leqslant \rangle$ 是一个格。这样一来,既可以把格看作一种特殊的偏序集,又可以把格看作具有满足交换律、结合律和吸收律的两个二元运算的代数结构。

如果 L' 是格 L 的非空子集合,对任意 $a,b \in L'$ 皆有 $a \cup b \in L'$ 和 $a \cap b \in L'$,则称 L' 为 L 的一个子格。设 $L = \{1,2,3,6,12\}$ 且 $L' = \{1,2,3,12\}$,并定义半序 $\leqslant$ 为:对任意二正整数 a 和 b,$a \leqslant b$ 当且仅当 b 能被 a 整除。这时 L 和 L' 显然都是格。但 L' 不是 L 的子格,虽然有 $L' \subseteq L$。

一个包含格的元素和符号 $=, \leqslant, \geqslant, \cup, \cap$ 的命题的对偶命题,是指用 $\geqslant, \leqslant, \cap, \cup$ 分别代替该命题中的 $\leqslant, \geqslant, \cup, \cap$ 而获得的命题。格的**对偶原理**是:若命题 P 在任意格中都成立,则 P 的对偶命题也在任意格中成立。

如果格 L 中有元素 0 和 1,且对每个 $a \in L$ 皆有 $0 \leqslant a \leqslant 1$,则称 L 为有界格,0 和 1 分别称为 L 的上界和下界。

设 A 为任意集合,则 A 的幂集 $\mathscr{P}(A)$ 关于集合的并运算和交运算构成一个有界格,其中 $0 = \varnothing$ 且 $1 = A$。

设 L 为有界格且 $a \in L$。若 $b \in L$ 使 $a \cup b = 1$ 且 $a \cap b = 0$,则称 b 为 a 的一个**补元**,这时也称 a 有补元。当 a 为 b 的补元时,a 亦是 b 的补元。若格 L 的每个元素都有补元,则称 L 为**有补格**。格中一个元素的补元不一定唯一。若格 L 中每个元素均有唯一的补元,则称 L 为唯一有补格。

如果对格 L 中任三元素 a,b 和 c 皆有 $a \cap (b \cup c) = (a \cap b) \cup (a \cap c)$(此时自然也有 $a \cup (b \cap c) = (a \cup b) \cap (a \cup c)$),则称 L 为**分配格**。当分配格中的一个元素有补元时,其补元必是唯一的。

如果 L 为有穷集合,则称格 L 为有限格,否则称格 L 为无限格。

如果格 L 的每个子集合都有上确界和下确界,则称 L 为完备格。有限格显然是完备格。

有补分配格称为**布尔代数**。

设 L 为格。若对任意 $a,b,c \in L$,当 $a \leqslant c$ 时皆有 $a \cup (b \cap c) = (a \cup b) \cap c$,则称 L 为一个**模格**或**戴德金格**。分配格显然是模格。

参考文献

1. Bwrris S, Sanppanavar H. A course in universal algebra. New York:Springer-Verlag,1981

2. 王兵山,周贤林,王长英,何自强. 离散数学. 长沙:国防科技大学出版社,1985 (王水汀)

Geleibeiqi fanshi

格雷贝奇范式(Greibach normal form) 一种由 S. A. Greibach 提出的**上下文无关文法**的规范化形式。它对上下文无关文法的生成式形式加以某种限制。S. A. Greibach 证明了任意的上下文无关文法都存在一个等价的格雷贝奇范式。

若上下文无关文法 $G=(\Sigma,V,S,P)$ 中的生成式均为 $A\rightarrow a\alpha$ 的形式,$a\in\Sigma,A\in V,\alpha\in V^*$,则称 G 为格雷贝奇范式,缩写为 GNF。格雷贝奇在 1965 年证明了任何不含 ε 的上下文无关语言都可由一个格雷贝奇范式产生,这就是著名的格雷贝奇范式定理。M. C. Paull 和 D. J. Rosenkrantz 等人分别给出了将任意的上下文无关文法转换成等价的格雷贝奇范式的算法。

虽然对任意的上下文无关文法,都可进行简化,但根据格雷贝奇范式定理还可构造一个等价的格雷贝奇范式。这种范式不仅使相应的上下文无关文法的生成式异常简单而且规范。这对研究相应的上下文无关语言的结构很有意义,例如,由 GNF 产生的每个串,其长度与产生它的推导步数相等。

参考文献

Hpcroft J E, Ullman J D. Introduction to automata theory, languages, and Computation. Boston, MA: Addison-Wesley, 1979 (苏锦祥)

geren jisuanji

个人计算机(personal computer) 一种面向个人用户,具有独立、完整的计算和处理功能,易于搬动或携带的计算机。包括**工作站**、**台式计算机**、**笔记本计算机**、**上网本**、**平板计算机**、**个人数字助理**和可穿戴计算机都属于个人计算机的范畴。但是,现在主要以台式计算机和笔记本计算机为代表。

一般认为,1971 年由 John Blankenbaker 制作出来的"肯巴克 1 号"(Kenbak Ⅰ)是世界上第一台个人计算机。

1976 年,"个人计算机"一词才正式出现在《比特》杂志(Byte magazine)五月刊物上。1976 年 5 月,IBM 在日本发布了 IBM 5100 台式系统。1977 年 4 月,"苹果Ⅱ型"个人计算机出现在美国西海岸计算机展销会上。"苹果Ⅱ型"个人计算机有显示器、键盘,主机采用摩托罗拉 6502 芯片作中央处理器,系统可支持 BASIC、FORTRAN、COBOL 和 PASCAL 等高级语言。可以说,"苹果Ⅱ型"个人计算机是推动个人计算机发展的一个里程碑。

1981 年 8 月,IBM 公司发布 IBM 5150 个人计算机。IBM 5150 使用英特尔 8088 微处理器,微软 MS-DOS 操作系统。1983 年 3 月,IBM 在纽约发布 IBM 个人计算机 XT。20 世纪 80 年代,IBM 陆续推出以英特尔(Intel)的 x86 的硬件架构及微软公司的 MS-DOS 操作系统的个人计算机,并制定以 PC/AT 为 PC 的规格。苹果公司 1984 年推出了第一款具有图形用户界面的个人计算机 Macintosh。IBM 在 1984 年发布了 PCJr 和 PC-AT 两款个人计算机。

在个人计算机的发展过程中,Intel 公司的系列处理器芯片的进展以及微软操作系统的配合发挥了重要作用。例如:1982 年 2 月英特尔介绍了其 6 MHz 80286 微处理器;1986 年 8 月 80386 微处理器上市;1989 年 4 月发布 25 MHz i486 微处理器;1995 年 1 月 63 MHz 奔腾超速芯片(Pentium Overdrive chip)上市;1998 年 1 月介绍具有 MMX 指令的 333 MHz 奔腾Ⅱ(Pentium Ⅱ)微处理器;2000 年 1 月发布 600 MHz 移动奔腾Ⅲ(Pentium Ⅲ)处理器;2003 年 1 月发布下一代移动处理技术"讯驰"(Centrino);2007 年 1 月发表 2.4 GHz 酷睿 2(Core 2)双核处理器等。与此同时,微软往往也相应推出其新的操作系统。形成了一个长达 20 多年的微软公司和英特尔公司的事实商业联盟的统治个人计算机的时期。

个人计算机主要有以下类型:**台式计算机**、**笔记本计算机**、**上网本**、**平板计算机**、**个人数字助理**、**可穿戴式计算机**等。

个人计算机的硬件

机箱 个人计算机主要硬件的承载物,其中主要包括主板、**中央处理器**、**主存储器**、硬盘、**软磁盘驱动器**、显卡、网卡、电源、**显示器**、键盘和鼠标等。机箱设计主要考虑用户方便使用,易于搬动、放置以及部件易于安装和良好散热等。

主板(mainboard) 主板又称为母板(motherboard),安装在机箱中,一般上面有 BIOS(基本输入输出系统)芯片、I/O 控制芯片、键盘和面板控制开关接口、指示灯插接件、扩充插槽、主板及插卡的直

流电源供电接插件等元件。

中央处理器(central processing unit, CPU)　又称中央处理单元,简称**处理器**,是整个计算机系统中最重要的部件之一,控制整个计算机主要的算术逻辑单元(arithmetic logic unit,简称 ALU),使计算机程序和操作系统可在它上面运行,其性能直接影响计算机系统的工作效率。

主存储器(main memory)　通称**内存**,属于内部存储器的一种,是计算机的"短期临时存储器",用于存放当前正在运行的程序以及目前所需的数据等。它的读写速度要远远高于硬盘驱动器或者光盘驱动器这些大容量存储设备,但是当系统关闭或没有电源供应的时候它的存储内容就会丢失。

硬盘　用于存储数据的最重要和传统的部件之一。与内存不同,其中的数据在断电之后不会丢失。硬盘在使用时会经由马达带动具有铁磁性表面的盘片飞速旋转,经由磁头写入和读取数据。

软磁盘驱动器　简称**软驱**,是以往个人计算机的重要设备之一,可插入可移动式的软盘。2000 年前由于价格便宜,并且是当时系统恢复及升级 BIOS 时必备的工具,因而在大多数计算机上都有装备。然而软盘容量小,稳定性不佳,现时已几乎被闪存所制成的 USB 闪存盘取代,在新配置的个人计算机上已罕见软驱的身影。

显卡　是计算机显示图像的重要器件,它负责处理 CPU 传送来的图像指令和数据,并将处理后的结果输出至显示器,以最终形成图像。显卡可以分为独立显卡以及集成在主板芯片组上的集成显卡两类。

网卡　是个人计算机尤其是台式计算机和笔记本计算机与外界局域网连接的重要部件,通过网络连线和网络协议与其他计算设备联系。

电源　将家用交流电源转换为个人计算机硬件所需的直流电的部件。

显示器　又称监视器、屏幕等,是计算机的图像显示设备。对于台式计算机,显示器一般独立于机箱。显示器主要有 CRT 显示器和平板显示器(如液晶显示器)。

键盘　是计算机的重要输入设备之一。现代个人计算机标准键盘一般有 107 个按键,多沿用打字机所采用的 QWERTY 布局,只是新增了功能键、方向键等计算机所需的键,有的键盘还设有一些额外的功能键。键盘的每个键上均有标明其所对应的字母、数字或功能。

鼠标　是一个小巧、可滑动的设备,也是当代计算机重要和十分常见的输入设备之一。在图形用户界面上,用户通过滑动和单击鼠标等操作可向计算机输入命令和数据。鼠标可通过 USB 或 PS/2 接口与计算机相连,新式的鼠标还可以通过无线 USB 及蓝牙连接。

个人计算机的软件

操作系统　个人计算机最重要的软件之一,用于管理计算机资源,并可让用户通过各种程序使用这些资源。目前 PC 市场上占有率最高的是微软所推出的 Windows 系列,其次则是类 Unix 系列的操作系统,例如有开放式架构且逐渐窜起的 GNU/Linux,以及 FreeBSD 等。

应用软件　为了完成各种任务,计算机必须安装各种应用程序。应用程序要依赖于操作系统等系统软件的支持,系统软件同时也会为其提供内存管理、网络连接以及设备驱动等常规服务。常见的应用程序包括文字处理程序、媒体播放程序以及游戏软件等。

参考文献

1. Personal Computer. http://en. wikipedia. org/wiki/Personal_computer

2. Polsson K. Chronology of personal computers. http://pctimeline. info/index. htm

(侯紫峰)

geren ruanjian guocheng moxing

个人软件过程模型(personal software process model)　定义良好,可以指导软件工程师训练有素地工作的过程框架。于 2000 年由美国卡内基梅隆大学的软件工程研究所(SEI)提出,简称 PSP。PSP 提供的过程模型包含一组指导工程师计划、度量和管理个人工作的方法、表格和程序,适用于需求分析、测试、修复缺陷等大多数的软件开发工作。使用个人软件过程模型的目标是训练工程师可以在计划的时间和成本内,生产零缺陷产品,如果和团队软件过程配合使用,将更有助于达到这个目标。个人软件过程模型的基本思想基于以下原则:

(1)每一个工程师都是不同的个体,要达到有效工作的目标,他们必须制定个人的工作计划,而且最重要的是这个计划必须基于其个人的性能数据。

(2)要持续改进个人的性能,工程师必须使用结构良好并可度量的个人过程。

(3)要开发高质量的产品,工程师必须自觉

意识到他们对产品的质量有责任,并为之而努力。

(4) 缺陷风险的越早,需要的修复成本越低。

(5) 避免缺陷要比发现和修复缺陷更有效。

(6) 正确的方法永远是最快、最经济的方法。

PSP 模型主要有三个基本阶段,即计划、开发和事后总结,以保证以正确地方式工作。所以 PSP 要求工程师在做出承诺前,首先要做计划;在开发过程中,要度量花在每一个任务上的时间、引入和移除的缺陷、交付工作产品的规模等,任务结束时要分析工作的结果并利用发现的信息改进个人的过程。PSP 模型的过程流如图 1 所示。

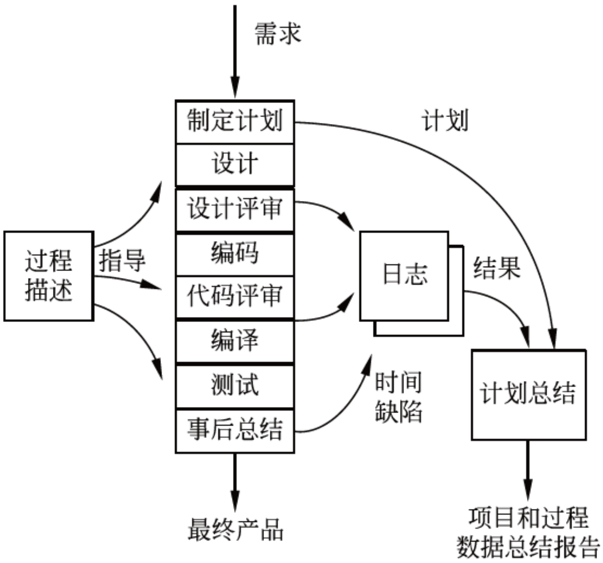


图 1 PSP 过程流

PSP 要求个人过程的改进也要循序渐进,图 2 表示 PSP 的过程改进演化。

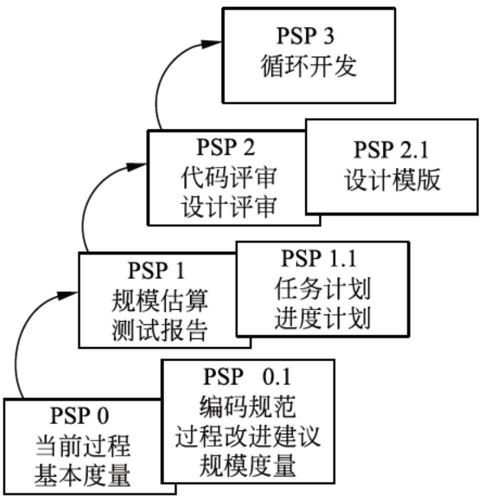


图 2 PSP 的过程改进演化

参考文献

Humphrey W S. PSP[SM]: a self-improvement process for software engineers. Addison Wesley, 2005 (王青)

geren shuzi zhuli

个人数字助理(personal digital assistant, PDA) 主要指满足个人信息管理需要的便携式信息处理设备。PDA 可以拿在掌上,装在衣服口袋里,用来帮助人们记录、处理商务和日常活动数据,俗称掌上电脑。PDA 与笔记本计算机相比具有轻便、小巧、耗电量低等特点。第一款 PDA 是 1992 年苹果计算机公司推出的 Newton。到目前为止最成功的 PDA 也出自苹果公司,但是形态已经有所改变,它是集成了通信功能的 iPhone 和更接近于电纸书的 iPad。在计算、通信、消费电子已经高度融合的今天,PDA 已经成为个人便携设备中必不可少的功能部分。

与传统**个人计算机**(包括台式计算机和**笔记本计算机**)相比,PDA 通常采用触控屏幕,支持使用触控笔或直接用手指进行输入和控制。为了与个人计算机中的个人事务数据保持一致,PDA 能够通过接口线或者无线连接与个人计算机进行同步。PDA 外形尺寸一般在 8 cm × 12 cm × 1 cm 左右,或者更小,质量一般在 100 ~ 350 g,或者更轻。PDA 的基本配置包括微处理器、存储器、显示器、键盘、接口、电源、系统软件及应用软件等。像个人计算机一样,操作系统是 PDA 的重要组成部分,也经常作为 PDA 分类的标志之一。PDA 所用的代表性的**操作系统**有 Palm OS、Symbian、Windows Mobile(前身为 Windows CE 和 Pocket PC)、iOS,以及安卓(Android)等。

PDA 的种类很多,分类方法也多种多样。一般可以从性能和用途进行分类。PDA 性能的外特性主要包括显示器的分辨率,使用电池的型号与电源消耗的快慢,数据交换能力,通信能力等;性能的内特性(系统特性),包括系统的软硬件配置和软件开放性(兼容性)等。系统配置包括微处理器的型号,内存的大小,各种接口等。PDA 的核心技术包括:缩小尺寸的硬件设计、省电的电源管理、嵌入式软件、数据的输入与输出等。

早期的个人数字助理产品一般包括:电子词典、电子记事本、手持计算机等。电子词典类的个人数字助理主要包括计算器、英汉汉英词典、电话号码簿、游戏软件等。此类设备的优点是价格便宜,体积小,重量轻,耗电量低;缺点是功能弱,可允许用户记

录的数据种类和数据长度有限。为了控制成本和耗电量,该类产品中使用的微处理器芯片一般是 8 位或 4 位的微处理器,显示器使用黑白和低分辨率的液晶显示器,电源使用 7 号电池或者纽扣电池,键盘使用与电子计算器类似的键盘。此类产品的功能和软件一般是固定的,不能升级。电子记事本类的个人数字助理的产生起源于人们希望用计算机代替笔记本,因此使用笔输入代替键盘输入成为此类设备的关键要求。此类设备的优点是体积小,重量轻,应用软件较丰富,耗电量较低;缺点是功能较弱,系统开放性不好,与个人计算机软件无法兼容,网络与通信能力较差。该类设备一般使用 16 位的微处理器芯片,显示器使用黑白 160×160 或 320×240 等较高分辨率的液晶显示器,电源使用 5 号或 7 号 AA 电池,输入使用触摸屏代替键盘。其软件大都具备基本的个人信息管理功能,主要包括电话本、备忘录、日程管理、个人理财、万年历、汇率转换、计算器、电子词典、口语练习、辅助写作、电子游戏等。此类产品一般具有和计算机进行数据交换、数据同步和应用程序交换的能力,软件一般可以升级。比较高级的电子记事本还具有收发电子邮件和传真等的功能。手持计算机的尺寸与电子记事本相近或略大一些。此类设备的核心设计思想是软件可以与笔记本计算机和台式计算机兼容,所以需要考虑使用通用操作系统或经过剪裁的通用操作系统。此类设备的特点是开放性好,软件丰富,功能强;缺点是耗电量较高,成本高。该类设备一般使用 32 位的微处理器芯片,显示器使用彩色高分辨率的液晶显示器,电源使用 5 号 AA 电池或专用电池,输入使用触摸屏代替键盘。这种计算机所支持的应用软件种类非常丰富,包括办公套件、游戏、网络应用、视频音频播放软件等。

面向行业的个人数字助理是一类重要的产品类型。例如商业领域具有条形码阅读、磁条识别、射频识别、智能卡阅读等功能的便携数字设备,军事领域的单兵数字助理,教育领域的便携学习机,公共安全领域支持人员信息管理乃至面像识别的个人信息助理,检验检测领域的便携数字检测仪,医疗健康领域的个人生理信号采集处理监测设备等。

随着**互联网**和移动通信技术的广泛应用,计算、通信、消费电子和内容已经在个人便携设备中高度融合,互联网接入、移动通信、视听功能、教育内容等相继加入,PDA 和手机、个人娱乐设备、学习机等之间的界限已经模糊,纯粹的 PDA 已经逐渐退出市场,或者说 PDA 的功能已经成为个人便携设备的一个必不可少的组成部分。近年来具有 PDA 功能的新型个人便携设备包括智能手机、电子图书阅读器、便携学习机等。智能手机除基本的移动通信功能外,已经具有 PDA 的各种功能,而且通过增加**全球定位系统**(GPS)、互联网接入、视听娱乐等功能,成为个人携带的第一选择。电子图书阅读器是便携式数字内容阅读装置,通常采用大容量移动存储卡存储大量图书,也有的支持网络下载或在线阅读,近期的电子图书阅读器采用基于电子墨水技术的非传统刷新式屏幕,能够提供不受户外强光影响、接近纸介质阅读的用户体验和极小的电源消耗。便携式学习机的主要特点是在 PDA 轻便携带的基础上,存储了大量的教育类资源,例如与课堂教学同步的辅导材料和音视频内容等,在中小学生和部分大学生中得到了广泛应用。近期,融合电脑、上网、娱乐、移动通信等功能的个人信息终端成为时尚热点,PDA 最初希望的人手一台的梦想正在成为现实,在掌上进行计算和网络通信的用户群正在迅速扩大,因此而带动的移动互联网服务和新型应用将对信息技术、信息产业乃至社会发展的方方面面产生重要而深远的影响。　　(高文　黄铁军)

geren wangluo

个人网络(personal area network, PAN)　一种用于在贴近人体的计算机设备之间进行通信的网络,也称个人区域网络。PAN 的覆盖范围通常只有几米,既可以用于进行设备自身之间的通信,也可以用于连接更高一级的网络或者互联网。个人区域网络可以通过 USB 或其他方式与计算机总线相连接,也可以通过无线网络技术进行组建,例如 IrDA、蓝牙、无线 USB、Z-Wave 以及 ZigBee。

无线个人区域网络(WPAN)是一种依靠无线网络技术连接的个人区域网络,通常无线个人区域网络都会采用短距离通信的网络技术,例如蓝牙以及由它所发展出来的 IEEE 标准——IEEE 802.15。

现如今 WPAN 已经得到了一定程度上的广泛应用,它可以用于将人们随身携带的传统计算以及通信设备互连起来,甚至用于较为特殊的应用,例如让外科医生在手术期间与其他的团队成员通信。

WPAN 技术中的一个关键的概念被称为“插入”,在理想场景下,当任意两个配备有 WPAN 的设备相互靠近或在一个中央服务器周围的时候,它们之间可以进行通信。另外一个重要的特性就是每个

设备都能够选择性的屏蔽其他设备,以防止不必要的干扰或者未授权的信息获取。

参考文献

1. Gutierrez J A, Naeve M, Callaway E, et al. IEEE 802.15.4: A developing standard for low-power low-cost wireless personal area networks. IEEE Network, 2001, 15(5): 12-19

2. Li J, Bose A, Zhao YQ. The study of wireless local area networks and wireless personal area networks. Canadian Conference on Electrical and Computer Engineering, 2005: 1415-1418

3. Zhao F, Viehland D. Key applications for success of personal area networks. Eighth International Conference on Mobile Business (ICMB 2009), 2009: 227-232

(崔勇)

gongcheng shujuku

工程数据库(engineering database) 为工程应用的特殊需要而产生的一种特种数据库,是存储、管理和使用工程设计所需数据的一种数据库系统,适合于CAD/CAM/CIM、地理信息处理、军事指挥、控制、通信等工程应用领域的数据处理。又称工程数据库为CAD数据库、设计数据库、技术数据库、设计自动化数据库等。

工程数据库处理的数据主要包括产品和零部件的图形和图像数据、产品文字数据、加工工艺数据(如加工设备、加工工艺规程、加工工艺、加工的数控代码等)、设计制造所需参数和设计分析数据(如设计标准、设备数据和材料数据),这些数据具有数据量大、数据类型和数据关系复杂、静态数据和动态数据相结合等特点。

由于工程数据类型和结构的复杂性,工程数据库需要提供表达和处理复杂对象的数据模型。工程数据管理的数据模型主要有非1NF数据模型、语义数据模型、面向对象数据模型以及对传统的网状、层次和关系模型进行扩充后的数据模型,相应的工程数据库语言主要包括扩充的NF2数据语言、自描述数据模型语言、基于网状结构的数据定义和操纵语言、面向对象数据语言。

工程数据库系统主要有两种实现方式:一种是在**关系数据库**系统的基础上加以扩充或改进;另一种是开发支持新数据模型的**数据库管理系统**。除了传统数据库的一般功能外,工程数据库还具有以下主要功能:

(1)多种实体表示形式 在设计和制造过程中,应用程序往往要利用同一实体的不同表示形式来实现不同的目的和要求。例如在几何造型中,可以使用CSC树、边界表示及八叉树法等多种方法来表示同一实体,该功能提供了存储和管理同一实体的多种表示形式的能力,并能保持这些表示形式之间数据的一致性。

(2)动态模式的修改和扩充 一个工程必须经过计划分析、设计、施工、调试、生产等阶段,相应的工程数据也是通过各阶段逐步明确和详细,最后才能得到满意的结果。产品的设计是一个变化频繁的动态过程,不仅数据变化频繁,而且数据结构也会有所改变,该功能提供了模式的动态修改及易于变化的能力。

(3)多版本管理 工程设计具有试探性、反复性和共享性,经常在同一对象的不同版本间进行切换。版本反映对象的演变过程。版本的表达方式根据实际应用背景的要求进行选择。多版本管理提供了版本生成、删除和切换的功能以及对复合对象的版本管理。

(4)多用户交互式工作 现代设计工作靠团队集体完成,以便设计人员之间了解彼此的思想和意图。该功能使团队中的设计人员可以采用并行工作的方式,他们既能同时工作,又可以共享资源,随时存取自己设计中需要的数据。

(5)图形数据处理。

(6)导航式查询 由于工程数据库具有数据量大且数据间关系复杂的特点,除一般查询功能外,工程数据库还提供导航式查询功能来提高查询的效率。导航式查询主要是针对复合对象的查询,它利用复合对象中对象的层次构成查询路径,查询的路径可以是很复杂的。查询过程犹如领航员导航一样。

(7)长事务处理 和商用事务相比,工程事务具有长期性、协作性和试探性。为避免商用事务处理中简单的等待、回滚等技术带来的低处理效率,工程数据库提供了多层事务、嵌套事务、保存点、最终一致性等长事务处理技术来支持工程事务的这些特点。

工程数据库是随着数据库应用的日益广泛深入而发展起来的。20世纪70年代,数据库开始应用于工程领域,如**计算机辅助设计**(CAD)、计算机集成制造系统(CIMS)等。1975年美国洛克希德公司的Eastman首先将工程设计方法与数据库技术相结

合描述了一个专用于 CAD 的工程数据库，随后各研究机构、大学和有关公司也开展了工程数据库技术的研究，推出了很多工程数据库系统。具有代表性的工程数据库系统有：IMDAS 集成制造数据库系统、TORNADO 系统、MLDB 系统、IPIP 系统、RMI 系统等。随着工程数据库应用的不断深入和发展，工程数据库技术正日臻完善，工程数据库技术的发展也将进一步促进数据库技术的发展。

参考文献

1. Cattell R G G. Object data management: object-oriented and extended relational database systems. Addison-Wesley, 1991

2. 李昭原，等. 数据库技术新进展. 2 版. 北京：清华大学出版社，2007

（周龙骧　王翰虎　寿宇澄　孙建伶）

gongye kongzhi jisuanji

工业控制计算机（industrial control computer）　一种针对工业现场恶劣环境设计，适用于工业过程实时监控的计算机。这是一种“流行”的狭义的定义，在这种定义下，工业控制计算机简称“工控机”，也被称为工业个人计算机（industry personal computer，IPC），广义地说，指用于工业自动控制的计算机，既包括专用机，也包括通用机；有小至仅带单片机的数显仪表，有大致上万信号点的多层次分布式控制系统。以下仅指狭义的工业控制计算机（工控机）。

工控机一般由主机、输入/输出（I/O）板卡、通信板卡和相关软件组成。

主机　主机在总体上与 PC 类似。主机的机箱一般分为机架式和壁挂式两种，采用标准工业尺寸，在散热、抗震、防腐蚀、电磁兼容（EMC）等方面有一些特殊的设计；主机的结构一般采用底板加插卡架构；主机的总线有很多种，主要有 PC 总线（ISA、VESA、PCI、PC104、PXI 等）、STD 总线、VME 总线等，目前使用最多的是 PCI 总线。

I/O 板卡　I/O 板卡主要通过标准的 PC 计算机总线（如 PCI 总线、USB 总线等）实现各种信号的输入、输出，一般包括（隔离/非隔离）开关量输入/输出板卡、模拟量输入/输出板卡等。

通信板卡　工控机不仅支持串行通信，通常还支持**以太网**。利用通信板卡，工控机既可以支持远程 I/O，也可以组建更为复杂的监控网络。

软件　工控机上的组态软件可以对主机和各类板卡进行各种详细的设置，也提供了相应的接口供应用性开发。大部分工控机生产厂商既提供自己的软件方案，也可以跟通用 HMI 软件连接。

工控机具有以下特点：

（1）可靠性高　工控机在工业现场连续运行，需要有足够长的平均无故障工作时间（MTBF）和较短的故障修复时间，以保证高运行效率。

提高可靠性的措施主要有：选用高质量的元器件并降额使用；抗干扰的电路设计和可靠的布线；使用优质的工业电源；采用看门狗电路；抗恶劣环境的系统结构设计和热设计；具有抗恶劣环境并适合工艺操作的人机操作界面；具有后备设计，有的进一步采用容错及冗余设计等。

（2）实时性强　工控机对生产过程进行实时控制与监测，因此要求它必须实时响应信号变化，并实时进行报警和处理。大部分工控机配置实时多任务操作系统。

（3）环境适应性强　工业现场的恶劣环境条件（如高温、低温、高湿度、多粉尘、强震动、含腐蚀性的气体、强电磁干扰等）要求工控机具有很强的环境适应能力，某些应用甚至需要采用隔爆的形式实现危险Ⅱ区的应用。

（4）扩展性好　工控机多采用无源底板加插卡结构，通过增加和更换卡件可以很容易地实现采样和控制信号的变更；工控机使用的组态和实时运行软件可以方便地改变控制算法和操作界面；多种通信协议和相关的兼容性设计使工控机之间、工控机和企业管理系统之间的数据交互很便捷。

（5）易维护　工控机及其板卡很多具备自诊断功能，能够自动分析和预测故障；友好的组态和操作界面使设置合适的控制算法、建立复杂的报表、数据的二次加工等都非常容易；很多板卡和模块可以实现带电插拔。

近年来，在工控机基础上，许多厂家提出 PAC（programmable automation controller）的概念，希望能够利用可编程控制器（PC）的开放性和 PC 上丰富的软件来实现原来可编程逻辑控制（programmable logic controller，PLC）的功能（参见**可编程控制器**）。

参考文献

1. 周泽魁. 控制仪表与计算机控制装置. 北京：化学工业出版社，2002

2. 厉玉鸣. 化工仪表及自动化. 4 版. 北京：化学工业出版社，2006

3. 林锦国. 过程控制：系统、仪表及装置. 南

京：东南大学出版社，2001 （王为民 黄文君）

gongzuoliu guanli xitong

工作流管理系统（workflow management system） 依照**应用软件系统**业务逻辑层的业务处理流程，对参与整个业务过程推进中的各种文件和任务实施控制、调度和管理的系统。

工作流技术起源于20世纪70年代末和80年代初，主要用于常规性批量事务的管理和集成，并在政府、金融、制造等**办公信息系统**和**管理信息系统**中获得成功应用。1993年成立工作流管理联盟的国际组织，随后颁布了工作流参考模型、接口规格和产品的互操作标准等，工作流管理系统得到长足发展。21世纪以来，随着**互联网**和**电子商务系统**、**电子政务系统**的发展，工作流管理系统与基于业务流程重组的管理学思想不断融合，并正演化为业务流程管理系统，其涉及的理论、开发方法和技术都得到进一步提升。

根据业务需求，业务处理过程中的所有参与者（称为活动）存在着一定的时序和逻辑关系，将所有活动依它们的时序和逻辑关系连接起来便构成工作流程。工作流管理系统的功能主要有：对工作流程及其每个活动进行定义和建模；在运行时对工作流程中的每个活动进行控制和调度，包括与相关客户及外部应用程序实施交互管理。

根据工作流参考模型，工作流管理系统以工作流执行服务为核心部件，以流程定义工具、客户端应用、调用应用、管理监控工具为基本部件。核心部件内含一个或多个分布式工作流引擎，用以提供流程实例的运行环境，并解释执行流程实例。核心部件通过交互使用接口同各个基本部件连接，实现工作流的定义、创建和运行管理。在基本部件中，流程定义工具负责业务流程的定义和管理，通常使用**可视化**的流程图方式实现复杂的流程定义；客户端应用部件用来接受和管理客户向工作流执行服务发出的交互请求；调用应用部件是为了协作完成一个流程实例的执行而设置的一个部件，用于管理工作流执行服务向外部应用程序发出的调用以及不同系统的工作流执行服务之间的交互；管理监控工具用于流程执行情况的监控以及组织机构与角色等数据的维护管理。

工作流管理系统能够实现更好的业务过程控制、修改和优化，从而可以有效提高业务流程的柔性，增强业务工作的适应性。目前，工作流管理系统与**面向服务的体系架构**、**业务智能**等技术融合，呈现出服务化和智能化的发展趋势，并正在加快向业务流程管理系统演化。

参考文献

1. Workflow Management Coalition. The Workflow Reference Model（WFMC-TC00-1003 Issue 1.1）. http://www.wfmc.org/standards/docs/tc003v11.pdf, 1995

2. Hollingsworth D. The workflow reference model: 10 years on. In: Layna Fischer, ed. The Workflow Handbook 2004. Future Strategies Inc., 2004

（刘江宁 吴泉源）

gongzuozhan

工作站（workstation） 一种性能高于**微型计算机**的多功能计算机。相比微型计算机，工作站在图形处理能力、数据存储能力和多任务并行处理能力方面进行了增强设计，为面向特定应用领域的人员提供一个具有友好人机界面的高效率工作平台。

工作站的多功能是指它的高速运算功能，多媒体应用功能和网络功能。高速运算包括**中央处理器**的高速定点、浮点运算以及高速图形和图像处理。多媒体应用是指工作站不仅能用于数值与文本数据处理，而且还能处理图形、图像、语音和声音。网络功能是指工作站在进行信息处理的过程中，可以通过网络与其他工作站或计算机互通信息和共享资源。

构成工作站的硬件有主机、**显示器**和输入输出设备。按21世纪10年代的技术水平，主机包括：运算速度在数百万（条）指令每秒（MIPS）的**中央处理器**；**主存储器**容量在数十GB；有多级**高速缓冲存储器**；磁盘容量从几百GB到几TB以上；机内设有各种总线接口、千兆**局域网**接口、音频和视频标准接口等，并且都采用高性能图形处理单元（GPU）显示卡增强图形处理能力。显示器具有高分辨率的彩色显示能力，用以实现高速图形显示、三维动态显示以及实时仿真功能。输入输出设备主要有**键盘**、**鼠标**和话筒等。此外，还可以接入磁盘、磁带、只读碟以及图形、图像输入输出用的外围设备，如数字化仪、彩色扫描仪、绘图仪、摄录设备以及其他音频视频设备等。

工作站的软件主要有：基本操作系统可以选择Linux系列、微软Windows系列和Apple操作系统系列；配备C，C++，FORTRAN等主要语言的编译器和

程序开发环境;基本的工具软件和支撑软件有数据库、图形和图像处理软件以及网络软件等;另外,还有相应的专业应用软件。

发展简史

工作站的发展大致可划分为5个阶段。

第一阶段为1973—1979年,是工作站实用化的研究开发阶段,典型产品有美国Xerox公司的Alto和MIT的CADR工作站。这个时期的工作站一般为:中央处理器采用16位**微处理器**,运算速度为0.2 MIPS左右;主存储器容量为128 KB至1 MB;显示器为单色,分辨率为800×600左右;网络为3 Mb/s以下的低速局域网;**操作系统**通常用UNIX。

第二阶段为1980—1984年,是工作站开始商品化并推广普及的阶段,典型的产品有Apollo公司的DOMAIN、Xerox公司的STAR和Sun Microsystems公司的SUN-1。这个时期的工作站一般采用16位微处理器(MC 68000,MC 68010等),运算速度为0.2~0.5 MIPS;主存储器容量为1~4 MB;显示器为单色,分辨率为1024×800;采用专用图形处理器;网络为1~10Mb/s的中速局域网;操作系统为UNIX或类UNIX。

第三阶段为1985—1986年,是工作站迅速发展时期,典型的产品有HP公司的HP 9000/320、Sun Microsystems公司的SUN-3以及SONY公司的NEWS。这个时期的工作站一般采用32位CISC微处理器(如MC 68020,MC 68030,Intel 80386等),运算速度达1~7MIPS;主存储器容量为4~32 MB;显示器为彩色(可256色),分辨率为1024×1024,64种灰度等级(单色时),采用二维专用高速图形处理器;网络为4~16 Mb/s的中速局域网;操作系统为具有网络功能的UNIX。

第四阶段为1987—2002年,是工作站鼎盛发展时期,工作站采用32位甚至64位精简指令集计算机(RISC)微处理器,不仅处理速度快,而且具有强大的图形、窗口功能和界面,并向多处理机、开放系统和分布式处理系统发展。典型的产品有SUN公司的SPARC系列和DEC公司的Alpha AXP工作站系列以及SGI、HP等公司的工作站系列。这个阶段的工作站一般采用32位或者64位RISC微处理器,运算速度达10~100 MIPS或以上;主存储器容量达16~256 MB;显示器为彩色(可2^{24}种色),分辨率为1280×1024,256种灰度等级(单色时),采用三维专用高速图形处理器;网络为4~16 Mb/s的中速局域网或者100~400 Mb/s的高速局域网;操作系统为可支持多处理机并行处理的UNIX SYSTEM V或OSF/1。

第五阶段为2002年至今,工作站进入与高端微型计算机融合发展时期。工作站采用64位多核x86系列微处理器,处理速度持续提高;采用高性能图形处理器GPU,具有强大的图形、窗口功能和界面。典型的产品有Apple公司的MAC Pro系列和惠普公司的Z8000工作站系列以及联想等公司的工作站系列。这个阶段的工作站一般采用64位x86多核微处理器,运算速度达8000~32 000 MIPS或以上;主存储器容量达4~192 GB;显示器分辨率为1280×1024以上,24位全彩色,采用专用高速图形处理器;网络为1000 Mb/s以上的高速局域网。

工作站的分类方法很多,按照工作站的用途可分为通用工作站和专用工作站。通用工作站没有特定的使用目的,可以在以程序开发为主的多种用途中使用。通常根据用途,在通用工作站上配置相应的硬件和软件以适应特殊用途。专用工作站是为特定用途开发的,由相应用途的硬件和软件构成,可分为办公工作站、工程工作站和人工智能工作站等。办公工作站是为了高效地进行办公业务,如文件和图形的制作、编辑、打印、处理、检索、维护,电子邮件和日程管理等;工程工作站是以开发、研究为主要用途而设计的,大多具有高速运算能力和强化了的图形功能,是计算机辅助设计、制造、测试、排版、印刷等领域用得最多的工作站。

工作站的应用领域十分广泛,例如:科学和工程计算、软件开发、计算机辅助分析、**计算机辅助制造**、**计算机辅助工程**、工程设计和应用、图形和图像处理、办公、金融和商业事务处理、过程控制和信息管理系统等。

参考文献

张树增,等.工作站——一种新型的计算机.北京:电子工业出版社,1992　　(张学孝　李英敏)

gonggong guanli xinxi xieyi

公共管理信息协议(common management information protocol, CMIP)　国际标准化组织ISO为解决不同厂商、不同设备的网络之间互通而制定的开放系统互连网络管理协议。该协议规范由ISO 9596(Information Processing Systems-Open Systems Interconnection Management Information Protocol Specification—Part 2: Common Management Information Protocol)给出了定义。CMIP是构建于OSI参考

模型上的网络管理协议,OSI 所定义的协议通常包括两个部分,一部分是指对上层用户提供的服务,一部分是指对等实体之间的信息传输协议。这里 CMIP 指实体间的信息传输协议,而对应服务则由 ISO 9595(Information Processing Systems—Open Systems Interconnection, Management Information Service Definition—Part 2: Common Management Information Service)所定义,称之为公共管理信息服务(CMIS)。CMIS 定义了获取和控制网络对象或设备的服务以及接收来自这些对象或设备的状态报告的服务。CMIP 与 CMIS 支持网络管理应用程序和管理代理之间的信息交换。CMIS 定义了每个网络组成部件需提供的网络管理服务,CMIP 则是实现 CIMS 服务的协议。

CMIP 采用 ISO 可靠的、面向连接的传输机制,具有内在的安全特性,可支持访问控制、授权与安全日志等功能。CMIP 并不指定网络管理应用的功能,仅定义被管理对象的信息交换机制,而不是信息如何被使用或被解释。CMIP 的协议结构如图 1 所示,其中会话控制服务元(association control service element, ACSE)用于建立和释放应用实体间的会话。远程操作服务元(remote operation service element, ROSE)是 ISO 中的远程过程调用。公共管理信息服务元(common management information service element, CMISE)是提供基本管理服务的服务元。

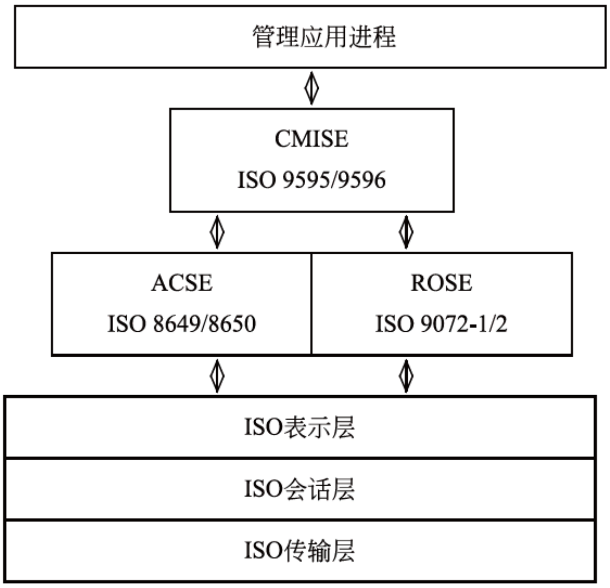


图 1　CMIP 的协议结构

CMIS/CMIP 在网络管理中的具体实现由于其复杂性和实现代价太高而遇到了困难。虽然 IETF 曾考虑将 CMIS/CMIP 应用于 TCP/IP 网络中,即 CMOT(The Common Management Information Services and Protocol over TCP/IP, CMOT),但作为一个临时性的过渡方案和在 SNMP 得到广泛认可的情况下,CMOT 已难有发展的空间。

参考文献

1. Sean Harnedy. 简单网络管理协议教程. 2 版. 胡谷雨,等译. 北京:电子工业出版社, 1999

2. Warrier U, Besaw L. The common management information services and protocol over TCP/IP (CMOT). RFC 1095, April 1989

（马皓）

gongli yuyi

公理语义（axiomatic semantics） 运用数学中的公理化方法给出的计算机语言的语义。不同的人在了解程序的含义时有不同的要求。例如,有的人只关心程序的数据输入和输出,而不关心程序是否正确终止。公理语义就是研究如何将这些不同的要求形式化,并根据这些要求严格给出程序设计语言的语义。

1967 年美国 R. W. Floyd 提出描述人们所关心的程序含义,以及如何去论证一个程序是否具有某种含义的数学方法,1969 年英国 C. A. R. Hoare 首次用公理系统定义了一类程序设计语言的语义。1975 年荷兰 E. W. Dijkstra 提出基于最弱前置条件的公理语义描述方法。

在定义语言的公理语义时,必须先给出描述所关心的程序语义的形式化方法,然后建立公理系统,规定语言成分的有关语义。如果用一个程序 P 去计算自然数的阶乘,这个程序中的变量 x 在程序开始执行时,存放用户输入的自然数值 k;而在程序执行终止时,存放要输出的结果。用户关心的是程序 P 计算的结果值是否确是输入值的阶乘。在公理语义中,使用公式 $\{x=k\}P\{x=k!\}$ 表示程序 P 的这一部分含义;若 P 执行前 x 的值等于 k,则 P 执行完毕后 x 的值等于 $k!$。程序 P 执行前的条件 $\{x=k\}$ 称为 P 的前置条件,执行后的条件 $\{x=k!\}$ 称为 P 的后置条件。这类公式称为归纳命题。一般地说,归纳命题用 $\{R\}P\{Q\}$ 表示;若程序 P 执行前,其程序变量的值满足前置条件 R,则程序 P 执行完毕后,其程序变量的值满足后置条件 Q。归纳命题用来作为描述程序语义的工具,公理语义就是用归纳命题的公理系统来定义程序语言的语义。

执行赋值语句 $(x:=e)$ 的结果是将程序变量 x

的值变为执行该语句前表达式 e 的值。也就是说,执行该语句后 x 的值等于执行该语句前表达式 e 的值。因此,若表达式 e 在语句 $(x := e)$ 执行前满足条件 R,那么程序变量 x 在语句 $(x := e)$ 执行完毕后亦应满足条件 R。故归纳命题 $\{R[e/x]\}x := e\{R\}$ 应该永远成立。其中 $R[e/x]$ 成立,表示将 R 中的 x 代为 e 后,R 成立,即 e 满足 R。在公理系统中,公理是一种永远成立的命题。这样采用 $\{R[e/x]\}\ x := e\{R\}$ 作为公理,就表达了语句 $(x := e)$ 的语义,使用不同的 R,可反映不同的用户对赋值语句语义的要求,R 可以只涉及输入输出变量;也可以涉及有可能产生副作用的其他变量。

执行顺序语句 $(s_1;s_2)$,就是使用执行 $(s_1;s_2)$ 前程序变量的值,先执行 s_1;然后使用执行 s_1 后程序变量的结果值,再执行语句 s_2,s_2 执行完毕后的程序变量值,就是执行顺序语句 $(s_1;s_2)$ 的结果值。故若执行 $(s_1;s_2)$ 前的程序变量值满足条件 R,$(s_1;s_2)$ 执行完毕后的值满足 T,那么就可找到关于中间结果的条件 Q,使得若 R 为 s_1 的前置条件,则 Q 为 s_1 的后置条件,而且以 Q 为 s_2 的前置条件,则 T 就是 s_2 的后置条件,这样就可以采用推理规则

$$\frac{\{R\}s_1\{Q\},\{Q\}s_2\{T\}}{\{R\}(s_1;s_2)\{T\}}$$

来规定顺序语句的语义。公理系统中的推理规则表示当横线上方的命题都成立时,则横线下方的命题亦成立。人们可使用不同的 R,T 来表示自己所了解的有关语义。

其他语言成分的公理语义也是用公理和推理规则类似地给出,但有的成分(如过程调用等)的语义,比起上述语句的语义要复杂得多。

论证一个程序是否具有某种含义的过程和论证一个程序是否具有某种特性的过程是完全一致的,故公理语义学是程序正确性研究的理论基础。程序验证的研究也进一步促进公理语义学的发展。

寻求适用于描述程序语义,且便于语义推导的逻辑语言是公理语义学研究的一个重要方面。20 世纪 70年代出现了使用时态逻辑来定义语言的语义,称为时态语义。另外如动态逻辑、算法逻辑在语义学中的应用,也都在发展之中。

参考文献

周巢尘. 形式语义学引论. 长沙:湖南科技出版社,1985

（李晓山　周巢尘）

gongneng guiyue yuyan

功能规约语言(functional specification language)　参见**功能语言**。

gongneng yuyan

功能语言(funcfional language)　用以书写软件功能规约的语言。软件功能规约是软件所要完成功能的精确而完整的陈述,它描述的是软件要做什么以及只做什么。功能规约是需求定义的功能抽象,它对需求定义中用户所需的功能进行重新组织和分块,使其表述为(接近于)数学模型的形式,以此作为软件设计与实现的依据。功能语言通常又称为**功能规约语言**。

在软件工程发展早期,软件功能主要采用非形式的自然语言加以描述。这样的非形式功能规约具有易书写、易理解和易使用的特点,一般用户均会使用;但由于自然语言的歧义性、模糊性和不完备性而导致难以开展软件的形式化和自动化方法的研究。为了提高软件生产率与软件产品可靠性,出现了形式软件功能规约语言。这种语言具有良好的数学基础,易于研究软件规约的各种性质和相互间的语义关系,如一致性、完备性、等价性和精化关系等,不仅避免了自然语言的歧义性、模糊性和不完备性,而且奠定了能保证程序正确性的各种形式化方法如 WP 和 VDM 等方法的基础。不仅如此,形式功能规约语言还使得软件自动化的实现成为可能。因此,形式功能规约语言逐渐为人们所接受并应用于软件工程实践。

从形式化的角度看,功能语言可分为非形式规约语言和形式规约语言。非形式规约语言是指未加限定的自然语言,而形式规约语言是指其语法和语义均显式精确定义的语言。由于软件功能的形式化描述是计算机科学中十分重要且相对成熟的领域,下文将集中讨论形式功能规约语言。从理论基础的角度看,功能规约语言可分为代数类语言和逻辑类语言。代数类语言系指以异调代数、范畴论等代数理论为主要理论基础的规约语言,如 OBJ, CLEAR 等;逻辑类语言系指以一阶谓词演算等逻辑理论为主要理论基础的规约语言,如 Z, VDM-SL 等。当然,有些语言将这两个途径有机地结合起来,例如,Larch 语言族中每一语言包括共享语言和接口语言两部分,其中共享语言基于代数方法,接口语言则基于逻辑方法。此外,还有一类广谱语言,广谱语言中

不仅包含功能规约机制，还含有设计规约等较低级的成分。

功能语言主要涉及规约对象、规约方法以及规约性质等。

规约对象主要包括抽象数据(数据抽象)和抽象过程(过程抽象)两类，它们分别反映了对软件系统中所涉及的数据以及相应的处理进行抽象的结果。抽象数据区别于数据的本质特征在于它只涉及数据及其运算的性质而与具体实现无关；而抽象过程区别于过程的本质特征在于它只刻画相应处理“做什么”的外部行为而与内部的实现细节无关。抽象过程是指从输入值集到输出值集的映射；抽象过程规约的主要问题是如何刻画抽象过程“做什么”的功能而不涉及“如何做”的具体算法与实现功效。抽象数据通常由抽象数据类型来反映。抽象数据类型是封装原理和信息隐蔽原理的集中体现；其本质特征在于与数据具体表示无关。因此，抽象数据类型规约的关键问题是如何体现这一特征。

规约方法主要研究如何对抽象过程与抽象数据类型进行规约。**前后断言方法**是一种常用的对抽象过程进行规约的方法。它通过给出一对基于一阶谓词演算的断言，来刻画抽象过程输入与输出的性质及其相互间的关系，以此来描述抽象过程的功能而与具体算法无关。采用前后断言方法，抽象过程的功能规约由两部分组成，其一是接口描述，它刻画了抽象过程与外界的接口，主要包括抽象过程的名、输入、输出以及输入输出的取值范围；其二是功能描述，它刻画了抽象过程“做什么”的功能，主要包括前断言和后断言两部分。一般说来，前后断言方法具有良好的数学基础，易于研究规约的性质和相互间的关系，如规约的一致性、完备性和精化关系等。但是，前后断言方法也有局限性，它受限于所能使用的谓词。

抽象数据类型规约的主要方法有二：其一是代数方法，即用等式公理来直接刻画抽象数据类型中运算的性质而与具体表示无关；其二是模型方法，即通过给出抽象数据类型的抽象模型来达到规约的目的。采用代数方法，抽象数据类型的规约由两部分组成，一是语法部分，二是公理部分。语法部分给出了抽象数据类型的名以及它所提供的操作的定义域和值域，同时，它也给出了与之相关的其他类型名；公理部分则通过给出一组刻画各操作性质的方程作为公理来定义各操作的含义。从语义的角度看，代数规约的语义模型有多个且形成一个谱，谱的两端分别称为始语义模型和终语义模型，从而为相应的实现提供了灵活性。归结起来，抽象数据类型的代数方法能够完整地刻画其运算的性质而与表示无关，从而较好地体现了抽象数据类型的特点。然而，它也有一定的局限性，对于某些简单的数据类型.要给出其代数规约十分困难。一个著名的例子是 M J Majster 提出的遍历栈。对于这样一个简单的数据类型，如果要使用代数方法，则要么使用隐含的辅助函数，要么使用无穷多的公理，结果均难令人满意。

与代数方法不同，模型方法不是对抽象数据类型中运算的性质加以直接刻画，而是通过某些已知的具有良好数学性质的(抽象)数据类型来给出所要定义的新类型的抽象模型，用已知类型运算的性质来间接刻画要定义的数据类型中运算的特性，从而达到定义新的抽象数据类型的目的。一般说来，抽象数据类型的模型规约由以下三部分组成：

(1) 状态集的定义(可能包含不变式)；

(2) 初始状态定义(通常只有一个)；

(3) 运算集的定义，通常采用输入输出断言刻画。

模型只能理解成行为的描述而与具体实现无关；但如果不加注意，模型规约可能引入与实现有关的内容。这种现象称为实现斜偏。

由于形式规约语言具有较好的数学基础，因此，易于研究由它所书写的形式规约的性质和相互关系，从而为形式规约的正确性验证和软件的形式化与自动化开发奠定了良好的基础。一般说来，**规约性质**包括一致性、完备性、等价性、精化关系等，其中一致性和完备性是为了保证软件规约在某种意义下是正确的；而等价性和精化关系通常用以保证软件开发过程的正确性。同一类性质在不同类型的规约结构中常常有不同的含义与表现形式。例如，就完备性而言，它在抽象过程功能规约中的含义不同于它在抽象数据类型代数规约中的含义。

概括起来，功能语言(特别是形式化功能规约语言)的研究取得了较大进展，其理论基础日臻完善，方法日趋成熟，并逐步应用于软件工程实践。进一步的工作包括它在大型软件开发中的应用以及基于形式化功能规约语言的形式化与自动化方法的研究。

参考文献

1. 徐家福，陈道蓄，吕建，等. 软件自动化. 北京：清华大学出版社，南宁：广西科学技术出版社. 1994

2. Gehani N, McGettrick A. Software specification techniques. Addison-wesley Publishing Company. 1986　　（吕建）

gongying guocheng

供应过程（supply process）　供应者为获取者提供软件产品的一系列活动。它从理解系统或软件产品的需求开始，经过准备投标、签定合同、制订计划、实施和控制、评审和评价等活动，直至交付完成。供应者是那些提供软件产品的机构。

供应者要对供应过程进行管理（参见**管理过程**）；并结合项目对供应过程的各项活动进行剪裁（参见**剪裁过程**）。

供应过程主要包括以下各项活动：

（1）开始　供应者要对项目招标书中的系统需求等进行认真研究，并结合自身的方针和其他因素对照分析，决定是否投标或接受合同。

（2）准备投标　为响应招标作好准备并编写一份投标书。

（3）签订合同　供应者就向获取者提供软件产品与他们进行有关合同的谈判，并签订合同；供应者可以请求修改合同，并以此作为改变控制机制的一部分。

（4）制定计划　该活动有下列各项具体任务：①供应者要对合同中所规定的系统需求进行全面分析，从而确定一个机构，以用于管理和确保完成该项目，以及对要交付的软件产品进行质量保证。②如果在合同中未具体指明，那么供应者要选用与项目的范围、规模和复杂程度相匹配的软件生存周期的各有关过程和活动。③确定计划要求，其中最好要包括对资源的要求和获取者的适当参与等。④根据风险大小等因素，对开发软件产品方案进行选择，一般来说有以下几种选择方案：使用内部资源开发；以子合同方式开发；从内部或外部来源取得现货产品；上述三种方案的结合。⑤制定项目管理计划，并将其写成文档，文档中一般要包含如下内容，但不仅局限于这些：项目进行中每个组织机构（包括外部机构）的责任和权利；应用于开发、运行和维护的工程环境，其中包括测试环境、程序库、设备、仪器设施、标准、规程和工具等；选用的有关过程和活动的工作细目，其中包括软件产品、不交付的软件、文档、预算、人力物力资源、软件规模及任务进度等；软件质量管理和保证的要求，必要时要另行制定质量保证计划；系统安全、保密和其他关键需求的管理，必要时要另行制定安全和保密计划；如果要建立分包合同，则要对分包合同的供应者进行管理，其中包括对他们的选择以及他们和主合同获取者的参与等；质量保证（参见**支持过程**）；验证和确认（参见**支持过程**），一般还要包括与验证和确认机构的联系、结合等方法；按合同要求，由获取者参与的联合评审，审计（参见**支持过程**），以及和获取者的非正式会面、报告、修改及其实施、验收等；风险管理，即涉及潜在技术、成本和进度诸风险因素的管理；保密方针，即在有关结构层次上存取信息的准则和规定；管理上要求的批准、专有权利等；对计划、跟踪和报告的方法；人员培训。

（5）实施和控制　该活动中供应者要落实和执行以上制定的项目计划，并依照**开发过程**中有关活动来开发软件，依照**运作过程**中有关活动对其运作，依照**维护过程**中有关活动内容对该软件产品进行维护，在整个项目计划实施过程中，供应者应监督和控制软件产品的开发进展和质量，这是一个连续且反复执行的任务，具体内容为监督技术性能、成本、计划、项目开展情况报告；发现问题、记录和分析解决问题；若有分包合同，则要对分包合同的供应者进行管理（参见**管理过程**），并向他们传达必要的主合同要求，以保证交给主合同获取者的软件产品均能符合必要的主合同要求；供应者还要按照合同和项目计划的规定与独立的验证、确认、测试机构及其他有关各方进行交接和联系。

（6）评审和评价　在适当时机，供应者要按合同开展评审活动，并与获取者加强联系，如支持非正式会面、验收评审、验收测试、联系评审和审计等。另外，供应者还要进行软件产品的验证和确认，以表明软件产品满足系统需求等各项目要求，并向获取者提供关于评价、评审、测试和问题解决的报告，需要时还要进行质量保证等其他有关活动。

（7）交付和完成　按合同规定向获取者交付软件产品并就该产品向他们提供必要的支持。

参考文献

1. IEEE Standard for Developing Software Life Cycle Processes-IEEE Std 1074—1991

2. ISO /IEC 12207：1995 Information Technology—Software—Part 1： Software Life-Cycle Process. 1995　　（陈森芬）

gongxiang cunchu

共享存储（shared memory）　两个或多个处理

机共用一个**主存储器**的并行体系结构。每一个处理机都可以把信息存入主存储器,或从中取出信息。处理机之间的通信通过访问共享存储器来实现。

共享存储计算机系统由于支持传统的单地址编程空间,减轻了程序员的编程负担,因此它具有较强的通用性,且可以方便地移植现有的应用软件。早期的共享存储计算机采用集中式的共享存储器,即多个处理机通过总线、交叉开关或多级**互连网络**等与共享存储器相连,所有处理机访问存储器时都有相同的延迟。然而,由于多个处理机共享存储器,使得存储器成为系统瓶颈。为此,在规模较大的共享存储计算机系统中,把共享存储器分成许多模块,并使它们分布于各结点之间(一个结点可能有一个或多个处理机),这种系统称为**分布式共享存储**(DSM)计算机系统,即每个结点包含共享存储器的一部分,结点之间通过可伸缩性好的互连网络相连。分布式的存储器和可伸缩的互连网络增加了访存带宽,但导致不均匀的访存延迟。

为了缓解由存储共享引起的冲突以及由存储器分布引起的访存延迟,共享存储计算机系统的处理器中一般都有高速缓冲存储器。但高速缓冲存储器的使用带来了高速缓冲存储器一致性问题(参见**高速缓冲存储器一致性**)。另外,存储器的分布使处理机访问不同的存储单元有不同的延迟。为了保证并行程序的正确执行,需要有某种高速缓冲存储器一致性协议和存储一致性模型。

高速缓冲存储器一致性协议是把系统中的一个处理机新写的值传播给其他处理机的机制。高速缓冲存储器一致性协议都是为实现某种存储一致性模型而设计的。

存储一致性模型是系统设计者和程序员之间的一种约定,它给出了判断共享存储程序及结构正确的标准,其中顺序一致性模型被普遍作为共享存储程序执行正确的标准,也是定义其他弱一致性模型的基础。在顺序一致性模型中,多个处理器并行执行程序的结果等于把每个处理机所执行的指令流按某种方式顺序地交织在一起在单机上执行的结果。如果在多处理机环境下的一个并行执行的结果和同一程序在单处理机多进程环境下的执行结果相同,则此并行执行正确。

在共享存储系统中,为了实现顺序一致性模型,需要对访存事件次序施加严格的限制,为了放松对访存事件次序的限制,人们提出了一系列弱存储一致性模型。这些弱存储一致性模型的基本思想是:在顺序一致性模型中,虽然为了保证正确执行而对访存事件次序施加了严格的限制,但在大多数不会引起访存冲突的情况下,这些限制是多余的,因此,可以让程序员承担部分执行正确性的责任,即在程序中指出需要维护一致性的访存操作,系统只保证在用户指出的需要保持一致性的地方维护数据一致性,而对用户未加说明的部分,则可以不考虑处理机之间的数据相关。

根据存储器的分布、一致性的维护以及实现方式等特征,常见的共享存储系统的体系结构有以下几种:

(1)无高速缓冲存储器的集中式共享存储结构　这种结构的处理机没有高速缓冲存储器,多个处理机通过交叉开关或多级互连网络等直接访问共享存储器。由于任一存储单元在系统中只有一个备份,这类系统不存在**高速缓冲存储器一致性**问题,系统的可伸缩性受限于交叉开关或多级互连网络的带宽。采用这种结构的典型例子是并行向量机及一些大型机,如美国 Cray 公司的 Cray-XMP、YMP-C90 等。

(2)基于高速缓冲存储器的集中式共享存储结构　在这种结构的系统中,每个处理机都有高速缓冲存储器,多个处理机一般通过总线与存储器相连。每个处理机的高速缓冲存储器通过侦听总线来维持数据一致性。由于总线是独占性资源,这类系统的伸缩性是有限的。这种结构常见于采用对称式多处理机(SMP)系统(参见**并行处理系统**)的服务器和工作站中,如 4DEC,SUN,Sequent 以及 SGI 等公司的多机工作站产品。

(3)具有高速缓冲存储器一致性的分布式共享存储结构　这种结构称为高速缓冲存储器一致的非均匀存储访问(CC-NUMA)结构。这类系统的共享存储器分布于各结点之间。结点之间通过可伸缩性好的互连网络相连,每个处理机都能缓存共享单元,高速缓冲存储器一致性的维护是这类系统的关键,决定着系统的可伸缩性。常采用基于目录的方法来维持处理机之间的高速缓冲存储器一致性。这类系统的例子有 Standford 大学的 DASH 和 FLASH,MIT 的 Alewife,以及 SGI 的 Origin 2000 等。

(4)唯高速缓冲存储器的分布式共享存储结构(COMA)　在这种结构中,每个结点的存储器相当于一个大容量的高速缓冲存储器,数据一致性也在这一级维护。这种系统的共享存储器的地址是活动的。存储单元与物理地址分离,数据可以根据访存模式动态地在各结点的存储器间移动和复制。其优

点是当处理机的访问不在高速缓冲存储器命中时，在本地共享存储器命中率较高。其缺点是当处理机的访问不在本结点命中时，由于存储器的地址是活动的，需要一种机制来查找被访问单元的当前位置，因此延迟很大。采用这种结构的系统有美国 Kendall Square Research 公司的 KSR1 和瑞典计算机研究院的 DDM。此外，这种结构常用于共享虚拟存储系统中。

（5）无高速缓冲存储器一致性的分布式共享存储结构　这种结构称为无高速缓冲存储器一致性的非均匀存储访问（NCC-NUMA）结构。它的特点是虽然每个处理机都有高速缓冲存储器，但硬件不负责维护高速缓冲存储器一致性，而由编译器或程序员来维护。其典型代表是 Cray 公司的 T3D 及 T3E 系列产品，在 T3D 和 T3E 中，系统为用户提供了一些用于同步的库函数，便于用户通过设置临界区等手段来维护数据一致性。这样做的好处是系统伸缩性强，高档的 T3D 及 T3E 产品可达上千个处理机。

（6）共享虚拟存储结构　这种结构又称为软件分布式共享存储结构。它的基本思想是在基于消息传递（参见**并行处理系统**）的大规模并行处理系统或集群式计算系统中，用软件的方法把分布于各结点的多个独立编址的存储器组织成一个统一编址的共享存储空间。这种结构具有共享存储系统的可编程性和消息传递系统的硬件简单的优点，其主要问题是通信开销很大。在某些高性能计算机系统中，在结点内部利用对称多处理机系统提供硬件的共享存储，在结点间由软件实现共享存储。常见的虚拟共享存储系统有 Ivy，Midway，Munin，Treadmarks 和 JIAJIA 等。

参考文献

胡伟武. 共享存储系统结构. 北京：高等教育出版社，2001

（胡伟武）

gouzao leixing

构造类型（composite type）　其值由更简单的值组合而成的类型。这些更简单的值所具有的类型称为“分量类型”或“元素类型”，它们可以是简单类型，也可以是构造类型。

现有的程序设计语言提供了多种构造数据结构的方法，如元组、记录、变体、联合、数组、集合、字符串、表、树、顺序文件等。事实上，所有这些类型是通过下述概念来构造的，它们是：

（1）笛卡儿积　可形式地表示为

$$S = T_1 \times T_2 \times \cdots \times T_n$$
$$= \{(x_1, x_2, \cdots, x_n) \mid x_i \in T_i, 1 \leqslant i \leqslant n\}$$

笛卡儿积用于构造元组和记录，如下列记录类型

record
　　I_1, : T_1;
　　…
　　I_n : T_n
end

其值集为 $T_1 \times \cdots \times T_n$。

（2）分离并集　我们用 $S + T$ 表示集合 S 和 T 的分离并集，每一个值或选自 S 或选自 T，此外，每一个值还附以标记以表明其选自哪个集合。可形式定义为：

$$S + T = \{\text{left } x \mid x \in S\} \cup \{\text{right } y \mid y \in T\}$$

即选自 S 的值标记以 left，选自 T 的值标记以 right。

分离并集用于构造变体记录和联合类型。如变体记录

record
case I : T **of**
L_1 : (I_1 : T_1);
…
L_n : (I_n : T_n)
end

其值集为 $T_1 + \cdots + T_n$。

（3）**映射**　一个映射 m 把集合 S 中的每一值映射到集合 T 中的一个值，记作 $m: S \rightarrow T$。

程序设计语言中的数组实质上表示一种有限映射，即从一个有限集合（数组的下标集）到数组的分量集合的映射，大部分程序设计语言仅使用整型（或其子域）作为数组的下标集，而 PASCAL 和 Ada 语言允许数组的下标集为任意离散简单类型。

例：下列数组类型

type window = **array** [0..511, 0..255] **of** 0..1

其值为 window = {0,...,511} × {0,...,255} →{0,1}，

若变量 W 说明为：

Var W : window

则 W[8,12] 取接到 W 的下标为（8,12）的分量。

映射在程序设计语言中还以函数抽象的形式出现，函数抽象通过一个算法来实现从 S 到 T 的映射，函数取 S 中的任意值计算出该值在 T 中的象，所以

S 不必是有限集。

例：PASCAL 中的标准函数 odd 实现一个从 Integer 到 Boolean 的映射，即：

$\{0\to false, \pm1\to true, \pm2\to false, \pm3\to true...\}$

（4）幂集　考虑一个值集 S，S 的所有子集的集合构成 S 的幂集，形式地记为

$$P(S)=\{s|s\subseteq S\}$$

幂集用于构造集合类型，在 PASCAL 语言中，集合类型定义具有如下形式

set of T；

该集合类型的值集即为 $P(T)$。PASCAL 语言提供了所有基本的集合运算。

例：**type** Color =（red，green，blue）

Hue = **Set of** Color

类型 Hue 的值集为 P（color），即{red，green，blue}的所有子集的集合。

（5）*递归类型*　其值由同样具有该递归类型的值组成的数据类型，递归类型是根据其自身来定义。

一般地，一个递归类型 T 的值集是由下列递归等式所定义的最小解：

$T=\cdots T\cdots$

递归类型用于定义动态数据结构。

参考文献

Watt D A. Programming language concepts and paradigms. Prentice Hall，1990　（金凌紫　陈涵生）

gutaipan

固态盘（solid state disk，SSD）　一种以易失性存储器（如同步动态随机存取存储器）或非易失性存储器（如**快闪存储器**）作为存储介质实现与**磁盘存储器**功能等效的数据存储设备。基于 NAND 闪存（参见**快闪存储器**）的固态硬盘由控制单元和固态存储单元（Flash 芯片）组成。与传统磁性硬盘相比，摒弃了机械寻址装置，而具有高速传率、低功耗、耐冲击、高可靠性而被认为是未来外部存储设备的新星。虽然固态盘中已没有可旋转的盘状结构，但由于使用的接口和呈现的逻辑视图与硬盘一致，这类存储器仍被称之为“盘”。

分类

固态硬盘的存储介质主要分为两种：易失性半导体存储介质 DRAM 和非易失性半导体存储介质 NAND Flash。

（1）基于易失性存储介质 DRAM 的固态盘　采用 DRAM 为存储介质，目前市场上应用较少，是非主流产品。此类固态盘兼容大部分操作系统的文件系统工具进行卷设置和管理，并采用高速数据接口光纤通道（FC）或标准 PCI 等连接主机和服务器。基于 DRAM 的固态盘读取或写入数据的时间只要 0.015 ms，工作状态下随机速度达到了 40 万次 I/OPS。此种固态盘反复读写的次数非常高，因而使用寿命很长，但由于其介质特性断电会丢失数据，故需要提供独立的电源来保持数据。

（2）基于非易失性存储介质 NAND 闪存的固态盘　采用闪存芯片作为存储介质，也是我们通常意义上说的 SSD。其接口形式丰富多样，包括 USB 接口的 U 盘，SATA Ⅱ 接口的便携机所用的硬盘，PCIe 接口的存储卡，CFAST 接口的微硬盘等。因数据在掉电时仍可长期保存，故能适应于各种环境。由于介质特性，采用单层位元（SLC）技术芯片的复写次数为 10 万次，而采用多层位元（MLC）技术芯片的复写次数仅有 1 万，为了有效地保证固态盘的寿命，控制器设计时都采用磨损均衡算法（wear leveling）使得写操作尽量均衡地分布到每个闪存芯片中，以保证固态盘整体的复写次数。因此，这种固态盘最适合于以读操作为主的应用。

固态盘的结构

图 1 为一种固态硬盘的内部结构图，左边为 8 片 NAND 闪存芯片，在固态硬盘内部，闪存芯片的组织采用多通道的处理技术，有效地提高数据读写的并行性。右边中间的芯片为固态盘控制器，采用现场可编程门阵列（FPGA）芯片，完成数据的读写操作、纠错码（ECC）校验、均衡算法以及建立对应的数据传输通道。右上方是同步动态随机存储器（SDRAM），最右边为串行 ATA（SATA）接口（参见**外存储设备接口**）。

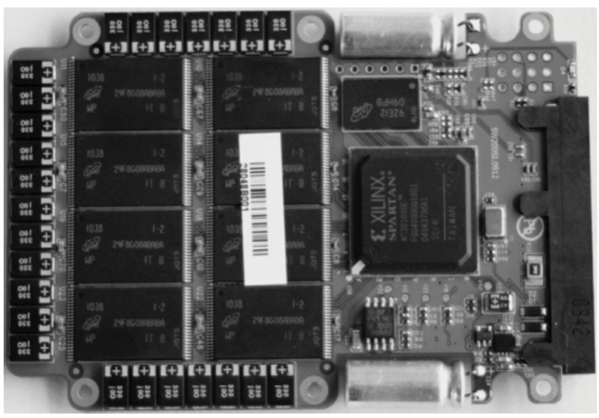


图 1　固态硬盘结构图

图 2 为固态盘控制器方框图，主要包含 3 大部

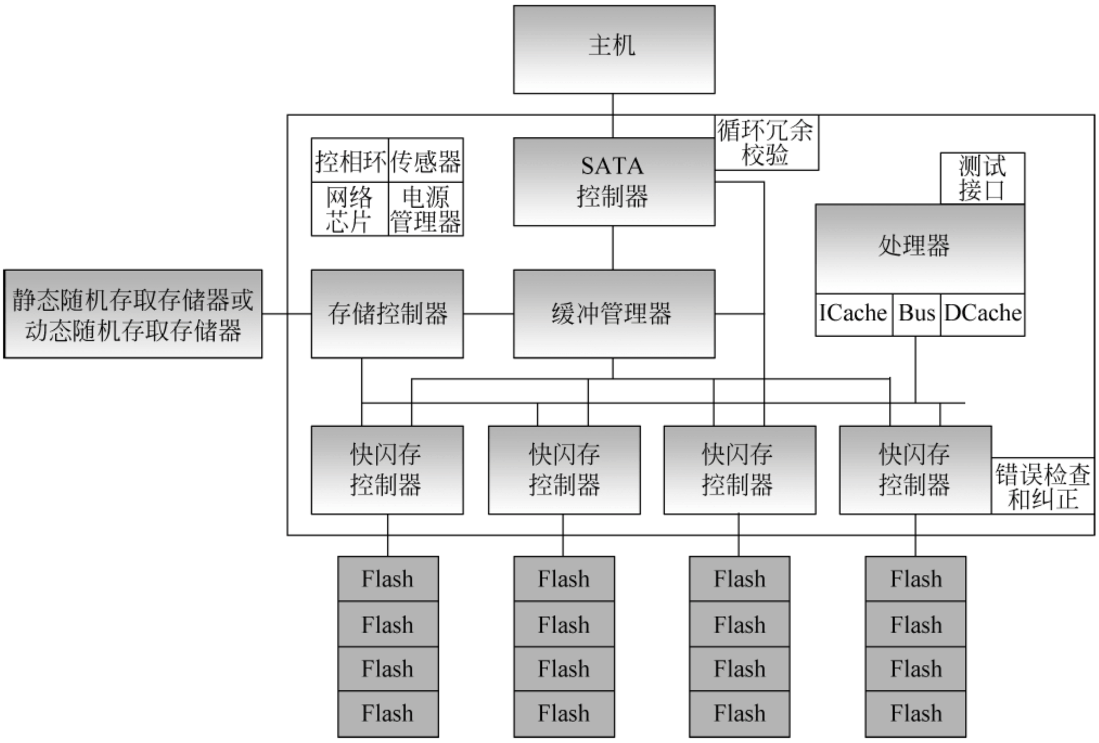


图 2　固态盘控制器方框图

分:外围接口、数据管理、Flash 控制器。SATA 控制器通过封装的 ATA 协议完成与主机进行数据交换,并将数据通过缓存管理器和控制器,暂存在同步动态随机存储器(SDRAM)中;处理器主要完成数据分配和数据管理功能;Flash 控制器负责向所连接的 NAND Flash 芯片执行读写操作。当执行写操作时,待写入的数据必须经过 ECC 校验之后,将数据和 ECC 一起写入 NAND Flash 芯片;但执行读操作时,数据会与其对应的 ECC 信息一起被读出并进行校验,校验后正确的信息才通过 SATA 接口将数据发送给主机,ECC 校验器位于 Flash 控制器内部。

固态盘特点

(1) 固态硬盘具有极快的连续和随机访问速度。磁性硬盘的响应时间主要由读写磁头寻道时间决定,通常是 ms 级,而且由于外道和内道的道密度不同,寻址速度也会受到影响。固态硬盘则是通过地址解码器直接寻址,通常是 μs 级,而且寻址速度与数据物理位置无关。

(2) 固态硬盘具有极低的能耗。由于磁性硬盘存在机械部分,即使没有数据读取而处于睡眠(sleeping)状态时的能耗也高于 10 W,其能效比很低。而固态盘因没有机械装置,通常情况下的单盘能耗仅为 1 ~ 2 W。根据 Intel 的研究,每消耗 1 W 的功率,固态硬盘的单位时间的 IO 次数达到 200I/O PS,而硬盘仅能达到 4.6 I/O PS。

(3) 固态硬盘的外形尺寸灵活多样,重量轻。由于机械部分的限制,磁性硬盘的外形尺寸往往是固定的 2.5 in 或 3.5 in 盘,其单盘重量都超过 700 g 以上,导致使用场景受限。而固态硬盘采用半导体芯片,可以适用于各种外形尺寸。单盘的重量仅为 20 ~ 30 g。

(4) 固态硬盘抗恶劣环境能力强。磁性硬盘由于磁头在高速旋转盘片上飞行(飞行高度多为 nm 量级),因而对于振动、冲击、温度、湿度都比较敏感。传统机械硬盘的工作温度为 5 ~ 50 ℃,导致在很多特殊环境下无法启动和正常工作;而固态盘内部不存在任何机械部件,不会发生机械故障,抗碰撞、冲击和振动的能力强。大多数固态硬盘可在 -10 ~ 70 ℃工作,一些工业级的固态硬盘还可在 -40 ~ 85 ℃下工作。

随着闪存技术的进步和产品价格的下降,固态盘得到广泛应用。随着人们对运作速度、可靠性和稳定性要求的提高,固态盘的市场占有率逐步提升。固态盘将广泛应用于移动终端、工控、视频监控、电力、医疗、导航设备等信息设备之中,还可由于其高性能在数据中心中发挥重要作用。

参考文献

Isard Birrell M, Thacker C, Wobber T. A design for high-performance flash disks. Operating Systems Review. 2007, 41(2): 88-93 （李之棠 吴非）

guzhang zhuru

故障注入(fault injection) 一种可靠性验证技术。通过向系统中刻意引入故障,提供一种模拟错误和测试恢复的方法。

故障注入技术要追溯到20世纪70年代,首次用在硬件级引入错误,这种类型的故障注入称为硬件故障注入(hardware implemented fault injection, HWIFI),后来发现通过软件手段也能实现错误的引入,并在软件系统评测方面发挥作用,这类技术被称为软件故障注入(software implemented fault injection,SWIFI)。

硬件故障注入技术用于模拟和测试系统内的硬件错误,最早这方面的实验无非是通过电路板连接短路并观察对系统的影响。如桥接故障,基本是作为硬件系统可靠性的一种测试方法,后来出现了专门的硬件来扩展这种技术。

软件故障注入技术包括编译时故障注入和运行时故障注入两种类型。

编译时故障注入是一种通过修改源代码向系统中注入模拟错误的技术。如突变检测,它改变某些正常的源代码使得它们包含错误。产生的错误和那些由程序员无意加入的错误非常类似。代码突变又称代码插入故障注入,它添加代码而不是修改现有代码,通常采用扰乱功能实现。

运行时故障注入技术采用软件触发器将故障注入到运行中的软件系统中。触发器可以有多种实现方式,例如基于时间的触发器(当到达规定时间时会产生一个中断,中断处理程序与定时器能够注入故障)、基于中断的触发器(在到达系统代码中的指定点或触发系统中一个特定事件时,采用硬件异常和软件陷阱机制产生一个中断)。另外还有一种针对协议软件的运行时故障注入方法,比如模糊测试(fuzzing test),它是一种通过向目标系统提供非预期的输入并监视异常结果来发现软件漏洞的方法,可用于协议软件的测试。

参考文献

1. Clark J A, Pradhan D K. Fault injection: a method for validating computing-system dependability. Computer, 1995: 47-56

2. Hsueh M, Tsai T K, Iyer R K. Fault injection techniques and tools. IEEE Computer, 1997, 30(4): 75-82 （邹德清）

guanjianzhen donghua

关键帧动画(keyframe animation) 通过在关键帧处对影响画面的参数设置指定值,然后采用插值技术自动计算出其余帧参数值的运动控制技术。关键帧的概念来源于传统的卡通片制作。在早期Walt Disney的制作室,熟练的动画师设计卡通片中的关键画面,亦即所谓的关键帧,然后由一般的动画师设计中间帧。在三维计算机动画中,中间帧的生成由计算机来完成,插值(线性插值或样条插值)代替了设计中间帧的动画师。所有影响画面图像的参数都可成为关键帧的参数,如位置、旋转角、纹理的参数等。

基于关键帧的插值运动控制技术的主要步骤包括: ①确定需控制的运动参数; ②设置 n 个关键帧参数; ③采用样条插值技术对 n 个插值点进行插值; ④对该插值样条进行离散采样,求得在某一帧时的参数值。

参考文献

Parent R. Computer animation: algorithms and techniques. 2nd ed. San Francisco, CA: Morgan Kaufmann, 2007 （金小刚）

guanxi daishu

关系代数(relational algebra) 研究关系及其运算的代数理论。

n 元关系是由某些 n 元序偶构成的集合。n 元关系的并、差运算的含义与通常的集合论相同。m 元关系 R 和 n 元关系 S 的笛卡儿积是形如 $\langle x_1,\cdots,x_m,y_1,\cdots,y_n\rangle$ 的序偶构成的集合,其中 $\langle x_1,\cdots,x_m\rangle\in R$, $\langle y_1,\cdots,y_n\rangle\in S$。$n$ 元关系 R 在分量 $i_1,\cdots,i_k(1\leq i_1,\cdots,i_k\leq n)$ 上的投影是形如 $\langle x_{i_1},\cdots,x_{i_k}\rangle$ 的序偶构成的集合,其中 $\langle x_1,\cdots,x_{j_1},\cdots x_{j_k},\cdots,x_n\rangle\in R$, $(j_1,\cdots,j_k)$ 是 $(i_1,\cdots,i_k)$ 的按序排列。关系可视为二维表格,因此可以给关系的每个列命名。给定关系 R 以及由常量、关系列名、算术比较运算子和逻辑联结词构成的公式 W, R 相对于 W 的选择是指 R 中所有满足 W 的序偶组成的集合。

E. F. Codd 已经证明:关系代数和关系演算在表达能力上等价。因此,关系代数和关系演算均可作

为评价关系数据库系统查询语言的标准。

基于上述原始的关系算子，可以定义关系的交、商和连接运算。这些派生算子表达能力强，并且直观自然，因此它们在查询语言中得到了广泛应用。

利用关系算子的性质，可以构造各种算法，对关系代数表达式进行优化，以提高关系查询的执行效率。由于关系代数表达式和关系演算表达式之间可相互转换，因此，这种算法不仅适用于基于关系代数的查询语言，也适用于基于关系演算或两者混合型的查询语言。

关系代数与关系演算共同构成关系数据库的主要理论基础。它在关系数据库的设计理论、查询语言的设计与查询优化等领域有重要应用。例如，目前广为使用的查询语言 SQL 就兼具关系代数和关系演算风格。

参考文献

Ullman J D. Principles of database systems. 2nd ed. Computer Science Press，1982

（谭庆平）

guanxi shujuku

关系数据库（relational database） 采用关系模型的数据库。

关系模型由关系数据结构、关系操作集合和关系完整性约束 3 部分组成。关系模型的数据结构非常简单，只包含单一的数据结构：关系。关系用二维表结构来表示各类实体及实体间的联系，二维表由行和列组成。一个关系数据库由多张二维表组成。

关系模型是建立在严格的数学概念基础上的。给定一组域（域是值的集合）D_1、D_2、…、D_n，这组域中可以有相同域，则其笛卡儿乘积 $D_1 \times D_2 \times \cdots \times D_n$ 的子集可以构成一张二维表，称为一个关系，也称作表。n 为关系的目或度。表中各列名必须唯一，称为属性名；唯一确定一个元组的属性组称为候选码；若一个关系有多个候选码，则选定其中一个为主码；一个元组中的某一属性值称为一个分量，关系的每一个分量，必须是不可分的数据项。

关系的描述称为**关系模式**。关系模式是一个五元组〈R，U，D，DOM，FD〉。其中，R 为关系名；U 为属性名；D 为一组域；DOM 为属性到域的映射；FD 为一组数据依赖，是一类完整性约束条件。某一时刻一个关系模式的实例称为关系状态，简称关系，也称基本关系。

视图是从一个或几个关系（或视图）导出的表。它与基本关系不同，是一个虚表。数据库中只存放视图的定义，而不存放视图对应的数据，其数据仍存放在导出视图的关系中。视图的作用主要是充当关系数据库的外模式，同时它也能提供一定程度的安全保护，并简化用户操作。

关系模型的操作部分把整个关系作为操作对象。具有关系处理能力的关系数据语言，可分为**关系代数**、**关系演算**和介于两者之间的语言。一般以关系代数作为度量语言处理功能的标准。关系代数除提供传统的集合运算如并、交、差运算外，还提供了选择、投影、连接等操作。如果由关系代数表达的任何查询均能为某种语言所表达，而不必使用迭代、递归命令，则可认为该语言具有关系处理能力。关系数据库的标准语言是 SQL 语言。

关系模型中有三类完整性约束：实体完整性、参照完整性和用户定义的完整性。实体完整性是指若一个或一组属性 A 是基本关系 R 的主属性，则 A 不能取空值。参照完整性定义了外码和主码之间的参照关系。如果 F 是基本关系 R 的一个或一组属性，但不是关系 R 的码，如果 F 的取值与基本关系 S 的主码相对应，则称 F 是 R 的**外码**（foreign key）。参照完整性规定外码 F 或者取空值（F 的每个属性值均为空值），或者等于基本关系 S 中某个元组的主码值。用户定义的完整性是应用相关的约束条件。

数据依赖和规范化理论是关系数据库研究的重要内容，数据依赖用来描述数据之间的一类重要的完整性约束条件，是语义范畴的概念；**规范化**的概念和**范式**的定义给出了判别关系模式好坏的准则，使数据库设计有了评价模式的理论依据，模式分解的概念和算法，又为数据库设计提供了辅助工具。

关系模型是由美国 IBM 公司 San Jose 研究室的研究员 E. F. Codd 于 1970 年发表的题为"A Relational Model of Data for Large Shared Data Banks（大型共享数据库数据的关系模型）"论文中首次提出的，开创了数据库关系方法和关系数据理论的研究，为关系数据库技术奠定了理论基础。由于 E. F. Codd 的杰出贡献，他于 1981 年获得了 ACM 图灵奖。

20 世纪 70 年代是关系数据库理论研究和开发原型的时代。其中以 IBM San Jose 实验室开发的 System R 和加州大学 Berkeley 分校研制的 INGRES 为典型代表。经过大量的高层次的研究和开发取得了一系列的成果，关系数据库从实验室走向了社会。20 世纪 80 年代以后，关系数据库管理系统日趋成熟并广泛应用，使数据库系统成为现代信息系统的基础设施和核心。

参考文献

1. 王珊，萨师煊. 数据库系统概论. 4版. 北京：高等教育出版社 2006

2. Date C J. An introduction to database system. 8th ed. Reading：Addison Wesley Publishing Company，2005

3. Ullman J D. Principles of database and knowledge base systems. Vol.Ⅰ，Ⅱ. Computer Science Press，1989

（王珊 陈红）

guanxi yansuan

关系演算（relational calculus） 一个声明性数据库查询语言。

当人们使用关系演算对**关系数据库**进行查询时，只用告诉数据库所需数据应该满足什么样的条件，而不必告诉数据库具体的查询步骤是什么。因此，关系演算被称为声明性查询语言。

到目前为止，数据库界提出了两种关系演算。一种称为*元组演算*，形式接近于数学中的谓词演算。另一种称为*域演算*，形式接近于数学中的一阶逻辑。经证明，两种关系演算同**关系代数**在表达能力上是等价的。基于这三种拥有共同表达能力的语言，数据库专业人员提出了“关系完备性”的概念。依照该概念，如果一种数据库查询语言具有以上三种语言中任何一种的表达能力，我们则称这种语言是“关系完备”的。

元组关系演算最初是由 Edgar F. Codd 同关系模型一起提出的。基于元组演算的一个查询可表示为

$\{t \mid \mathrm{COND}(t)\}$

其中 t 指代关系数据库中的一个元组；而 COND（t）指代 t 需要满足的公式，通常由原子谓词、逻辑算子（∧（与）、∨（或）和¬（非））、存在量词（∃）和全称量词（∀）组成。查询结果为所有满足 COND（t）的元组。假设一个数据库中有两张表（表即关系），分别为

员工（姓名，年龄，部门代号）；

部门（部门代号，部门名称）

那么以下的查询1用于查找年龄超过30岁的员工的姓名和年龄

查询1：$\{t.\text{姓名}, t.\text{年龄} \mid \text{员工}(t) \wedge t.\text{年龄} > 30\}$

而以下的查询2则是要查找拥有年龄超过60岁的员工的部门

查询2：$\{t.\text{部门名称} \mid \text{部门}(t) \wedge \exists s(\text{员工}(s) \wedge s.\text{年龄} > 60 \wedge s.\text{部门代号} = t.\text{部门代号})\}$

域关系演算是由 Michel Lacroix 和 Alain Pirotte 为关系模型提出的声明性查询语言。基于域演算的一个查询可表示为：

$$\{\langle X_1, X_2, \cdots, X_n\rangle \mid p(\langle X_1, X_2, \cdots, X_n\rangle)\}$$

其中 X_i 要么是一个域变量要么是一个常量；$p(\langle X_1, X_2, \cdots, X_n\rangle)$ 代表一个公式，通常由原子谓词、逻辑算子（∧（与）、∨（或）和¬（非））、存在量词（∃）和全称量词（∀）组成。查询结果为所有满足 p 的向量 $\langle X_1, X_2, \cdots, X_n\rangle$。例如，前面的查询1可用域演算表示为：（假设 A、B、C、D 分别表示姓名、年龄、部门代号、部门名称）

查询1：$\{\langle A, B\rangle \mid \langle A, B, C\rangle \in \text{员工} \wedge B > 30\}$

而前面的查询2可用域演算表示为

查询2：$\{\langle D\rangle \mid \exists A, C(\langle A, B, C\rangle \in \text{员工} \wedge \langle C, D\rangle \in \text{部门} \wedge B > 60)\}$

参考文献

1. Codd E F. A relational model of data for large shared data banks. Communications of the ACM，1970，13（6）：377-387

2. Codd E F. Relational completeness of data base sub-languages. In：Rustin R，ed，Data Base Systems. Prentice Hall，1972

3. Lacroix M，Pirotte A. Domain-oriented relational languages. In：Proceedings of VLDB，1977：370-378

（王珊 陈红 周烜）

guanli guocheng

管理过程（management process） 软件生存周期中管理者所负责的一系列活动。管理者负责对所从事的过程，例如，对获取、供应、开发和支持等过程的活动进行管理；软件管理过程适用于必须对各自的过程进行管理的任何一方。

软件管理过程的目的是在一定的时间和预算范围内，有效地利用人力、资源、技术和工具，完成预定的系统或软件产品，实现预定的功能和其他质量目标。管理技能往往是系统或软件产品成败和软件质量高低的重要条件。大型软件项目涉及较多的人力、资源和时间，管理问题尤为突出。

软件生产是一项劳动和智力密集的活动，它是以人为中心的过程，具有可见性差和定量化难的特点。可见性差是指软件研制进度不易标志，存在的问题不易及时发现和纠正，其过程容易出现修改和

反复。定量化难指软件的成本、生产率和质量不易度量。因此,软件管理有特殊的复杂性。

软件管理过程是随着软件工程的发展而发展的。早在 20 世纪 60 年代末,已经有人讨论大型软件研制项目的组织管理问题。软件工程实践中发生的种种问题,往往是与管理密切相关的,例如,进度推迟、经费超支、质量差、程序人员不称职等。可管理性成为软件生产工程化的重要标志。因此,与软件工程化、产品化过程相适应,形成新的分工,出现了组织管理软件生产的管理员。这些管理员的管理活动体现了最初的软件管理过程。

软件管理活动的研究与软件管理的研究密切相关。最初主要在人员和组织方面,如软件心理学的研究。随着软件技术的发展,软件管理自身出现了专门的方法、技术和工具来支持软件管理活动。

软件管理的对象是进度、系统规模和工作量估计、经费、组织和人员、风险、质量、作业、环境配置等。因此,软件管理一般可分为进度管理、成本管理、质量管理、人员管理、资源管理和标准化管理等。这些管理一般都有其各自的活动内容,软件管理过程把这些活动归纳起来,抽取活动共性,一般可包含下述活动:

(1) 实施本过程的准备 一开始先要搞清楚进行管理的过程的需求,管理者通过调查研究确认达到需求的可能性。在必要时可通过有关各方协商来修订和完善需求。

(2) 管理计划的制定 制定计划的任务包括规定进度、分配资源、决定与项目有关的组织和承担人员(包括人员的地位、作用、职责、规章制度等)、根据规模和工作量估计分配任务、风险定量化、制定质量管理指标、编制预算和成本、准备环境和基础设施等。

(3) 计划的实施和控制 管理者根据需求对过程进行控制。他们监督过程的实施,提供过程进展的内部报告和按合同规定向获取方提供外部报告,并应当调查、分析和解决在执行过程中发现的问题,对计划进行调整和修改。问题及其解决办法都应写成文档。

(4) 对计划完成程度的评审和评价 管理人员应对计划完成程度进行评审,对项目进行评价,并对计划和项目进行检查,使计划和项目在完成或变更之后保持完整性和一致性。

(5) 管理过程完成时编写文档 管理者根据合同决定此过程是否完成。如已完成,应从完整性方面检查项目完成的结果和记录,并把这些结果和记录编写成文档并存档。

参考文献

1. IEEE Standard for Developing Software Life Cycle Processes — IEEE Std 1074—1991

2. ISO/IEC 12207:1995 Information Technology — Software — Part 1: Software Life-Cycle Process. 1995

(刘光龙)

guanli xinxiku

管理信息库(management information base, MIB) 网络管理中数据的标准,取自 OSI/ISO 网络管理模型。管理信息是指被管对象必须保存的、可供管理程序存取访问的控制和状态信息,若干性质相近的被管对象的集合可构成被管对象组。所有的被管对象组合在一起构成的虚拟数据库称为 MIB,MIB 的定义与具体的网络管理协议无关。在网络管理中,管理程序就是使用 MIB 中这些信息的值对网络进行管理,如读取或重新设置这些值。

MIB 中的对象是由抽象语法记法(Abstract Syntax Notation One, ASN.1)的一个子集——管理信息结构(structure of management information, SMI)来定义的,其结构是层次化的,该结构又称为对象命名树(object naming tree)。图 1 所示为管理信息库的部分示例。

对象命名树的顶级对象有三个,即 ISO、ITU-T 和这两个组织的联合体。在 ISO 对象的下面有 4 个节点,其中标号为 3 的对象代表被 ISO 认可的成员组织。在其下面有一个表示为美国国防部(Department of Defense)的对象,其标号为 6。该对象的子节点代表 Internet 对象,标号为 1。当仅讨论 Internet 中的对象时,只需画出 Internet 对象以下的子树,见图中带阴影的虚线方框内容,并在 Internet 节点旁标注上{1.3.6.1}即可。Internet 对象节点的第二个子节点是 mgmt,其含义代表为网络管理,标号为 2。再往下的子节点是管理信息库对象,原先的节点名为 mib。1991 年定义了新的版本 MIB-Ⅱ,故节点名现改为 mib-2,其标识为{1.3.6.1.2.1},或{Internet(1).2.1},这种标识称为对象标识符。

最初的 mib 节点将其所管理的信息分为 8 个类别,每个类别下面有若干管理对象,总共约 114 个对象。该管理信息库最初是在 RFC 1066(Management Information Base for Network Management of TCP/IP-based internets)中发布的,称为“基于 TCP/IP 的 In-

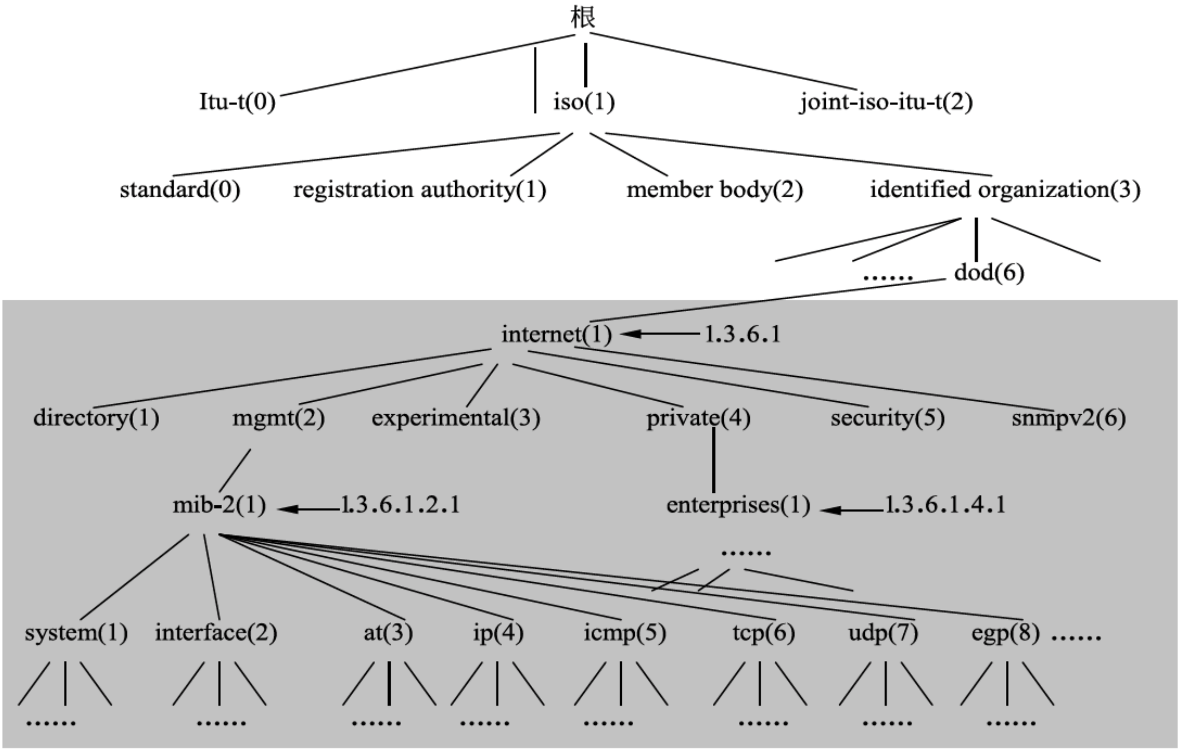


图 1 管理信息库的部分示例

ternet 的网络管理的管理信息库”，即 MIB-Ⅰ。这 8 个管理信息类别是：系统、接口、地址转换、IP、TCP、ICMP、UDP 和 EGP 等对象类，其中 IP、TCP、ICMP、UDP 和 EGP 属于 Internet 中广泛使用的 TCP/IP 协议集。1991 年发布的 RFC 1213(Management Information Base for Network Management of TCP/IP-based internets：MIB-Ⅱ)中又新增加了三个类别，成为 MIB-Ⅱ，现在 mib-2 所包含的信息类别已超过 40 个。

参考文献

1. Sean Harnedy. 简单网络管理协议教程. 2 版. 胡谷雨，等译. 北京：电子工业出版社，1999

2. Kurose J F，Ross K W. 计算机网络：自顶向下方法. 原书第 4 版. 陈鸣，译. 北京：机械工业出版社，2008

（马皓）

guanli xinxi xitong

管理信息系统(management information system，MIS) 利用人工过程、数学模型、**数据库**和其他计算机系统资源，通过数据的采集、传输、存储、加工和维护，为企、事业等社会组织的各种管理职能提供信息服务的一种**应用软件系统**。提供的信息包括数值数据、文字材料、统计图表及某些声像信息。管理信息系统(MIS)用以改善社会组织的运行效率及管理效果。

管理信息系统这一术语出现于 20 世纪 50 年代后期，但当时计算机刚引入管理领域，主要用来做例行的事务处理，如生产作业统计、进出账管理等，以提高事务处理的效率，减轻人工劳动。60 年代后期开始，随着**数据库**技术的出现和实用化，财务管理、人力资源管理、仓库管理、营销管理等管理系统不断完善，覆盖多个职能部门的综合数据处理系统也得到迅速发展，并逐步增加了分析、计划及控制等功能，用来支持决策过程，提高管理效果。70 年代出现的**决策支持系统**可以看作是 MIS 的发展和延伸，但 MIS 的决策支持功能通常是附带的，且涉及高层决策的功能主要交专门的决策支持系统实现。目前，MIS 已成为广泛应用和普及的应用软件系统之一。

任何一个 MIS 总是体现或支持某一种管理模式的。随着新的管理理念的出现，MIS 的内涵也在不断变化和发展。如面向制造业的管理信息系统，从基于定货点法的库存管理系统、支持物料需求的计划管理系统，到支持对所有制造资源进行统一计划和控制的管理系统以及在融合了多种管理理念的基础上，20 世纪 90 年代出现的对整个企业资源进行统一计划和管理的**企业资源规划**系统。

MIS 的逻辑结构与应用软件系统的逻辑结构是一致的,主要由计算机基础设施和应用软件组成。基础设施为应用软件提供运行环境等软硬件支撑。MIS 的功能由应用软件实现。应用软件的构造应与管理职能相适应,既可支持各种管理职能,也能支持每种职能不同层次上的管理活动。支持每一种管理职能都需要一组特定的数据和处理功能,它们便形成了 MIS 中各个相对独立的子系统。子系统之间借助分布计算中间件,通过通信集成、数据集成、应用集成、流程集成、门户集成和部门与部门之间的服务集成,消除信息孤岛,实现互联互通互操作和数据等信息资源的共享,使整个系统集成为一个整体。从管理活动的视角看,每种职能一般可以分为实现数据库查询与相应事务处理的运行层,实现管理业务基本需求的业务管理层以及辅助管理人员进行决策的战略规划层三个层次。通常,越底层的功能涉及的数据量越大,处理过程越确定,层次越往上,数据量越少,处理过程越复杂。以制造企业为例,反映以上概念的 MIS 概念结构如图 1 所示。

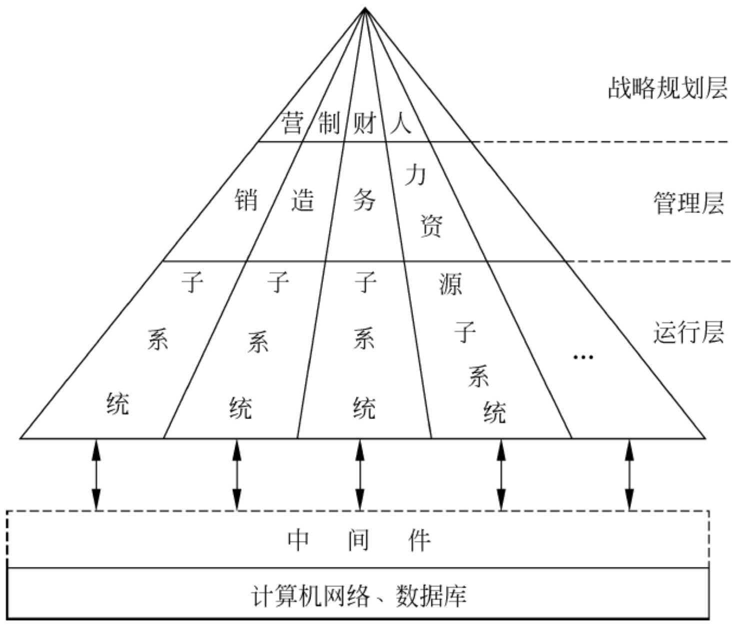


图 1　MIS 的概念结构

MIS 的开发主要是应用软件的开发,其开发方法和过程参见**软件工程**。开发完毕经测试验证后的系统可投入运行。主要有系统的日常运行管理,系统行为的合规性审查,系统功能与性能的检测与评估以及系统维护。

MIS 涉及的学科除计算机科学外,还有管理科学、运筹学、系统科学和行为科学等,可以说,系统的观点、数学的方法和计算机的应用是 MIS 的三要素。目前,MIS 与**决策支持系统**、**办公信息系统**、**电子政务系统**、**电子商务系统**和**企业资源规划**等系统正在不断融合,**云计算**模式将成为它们资源聚合和信息处理的发展趋势。

参考文献

1. [美]劳顿. 管理信息系统(原书第 11 版). 薛华成,译. 北京:机械工业出版社,2011

2. 仲秋雁. 管理信息系统. 北京:清华大学出版社,2010

(戴长华　吴泉源)

guang chuansong wang

光传送网(optical transport network,OTN)

由国际电信联盟标准化部门 ITU-T 的 G. 709、G. 872 等系列建议所规范,实现面向高带宽电路或波长级光层组网的新一代光传送与组网体系。

长期来,SDH 与 DWDM 是光传输网的基本形态,SDH 主要实现对 2 M/155 Mb/s 速率电路的组网和管理,而 DWDM 则主要完成点对点的大容量传输,缺乏相应的组网能力。随着业务容量的飞速增长,对更大颗粒度带宽(2.5 G/10 G/40 Gb/s 电路或波长)调度和组网的需求越来越大。在 1999 年,ITU-T 提出了面向光层组网的 OTN 概念,但由于全光网的实现一直面临诸多技术困难,OTN 也由专注光层管理和组网发展为包含高带宽电层管理和组

网。2001 年,ITU-T 正式发布了 G. 709 系列核心标准,形成了一个相对稳定的 OTN 体系。

如图 1 所示,OTN 光层分为光通道层(optical channel,OCh)、光复用段层(optical multiplex section,OMS)和光传输段层(optical transmission section,OTS)。

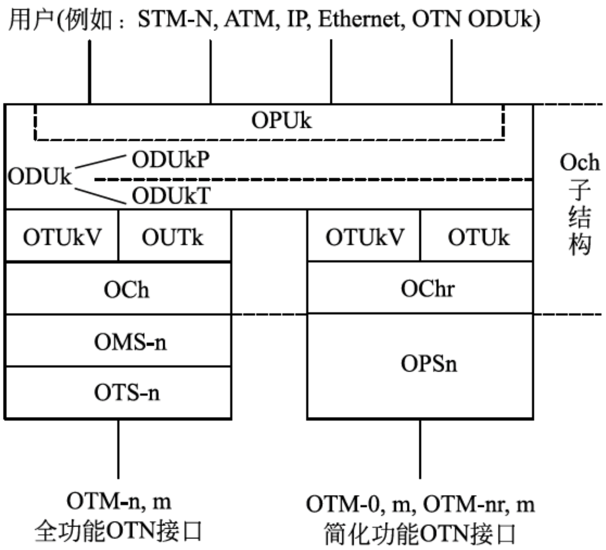


图 1　OTN 分层结构

数字包封(digital wrapper)技术是 G. 709 建议的重要内容,它定义了一种特殊的帧格式,将客户信号封装入帧后形成载荷单元,并在帧头提供用于运营、管理、监测和保护的开销字节,在帧尾提供前向纠错(FEC)字节。OTN 定义了多种映射方式,使不同类型的客户信号均有合适的路径适配进 OTN,OTN 还定义了虚级联帧,用来传送超过 ODUk 的大颗粒业务。

OTN 设备功能模型如图 2 所示,OTN 设备包括终端复用设备和交叉连接设备。终端复用设备就是支持 OTN G. 709 接口的 WDM 设备。交叉连接设备按照其交叉能力可以分为光层交叉、电层交叉、光电混合交叉三种,其中光层交叉基于波长或多个波长进行交叉;电层交叉具备子波长业务调度能力,还可以进行业务汇聚;光电混合交叉结合了上面两种交叉形式,通过光层交叉来调度大颗粒的波长级业务,通过电层交叉来调度小颗粒的子波长级业务,光电交叉互为补充。

OTN 支持多种保护方式,可根据不同的应用场景进行部署。在电层主要采用基于 ODUk 的子网连接保护(SNCP)、环网共享保护等,在光层主要采用光通道 1 +1 保护、光通道共享保护和光复用段 1 +1 保护等。另外还可以基于控制平面实现网络保护和业务自动恢复。OTN 提供了许多 OAM 功能,定义了丰富的开销字节,使其具备同 SDH 一样强大的运维管理能力。

OTN 可以应用到从长途骨干层到城域网汇聚层的多个层面。OTN 继承了 SDH 在组网、管理和保护等方面的优点,并能够提供更大颗粒度的传送和灵活的交叉调度,能够支持多种客户层协议,具备多业务的承载和适配能力,代表了光传送网的发展趋势。目前全球各大电信运营商已开始积极采用 OTN 进行组网,随着标准的成熟和业务的发展,OTN 正在得到越来越广泛的应用,传统的 IP over WDM 组网也将向 IP over OTN 组网方向演进。

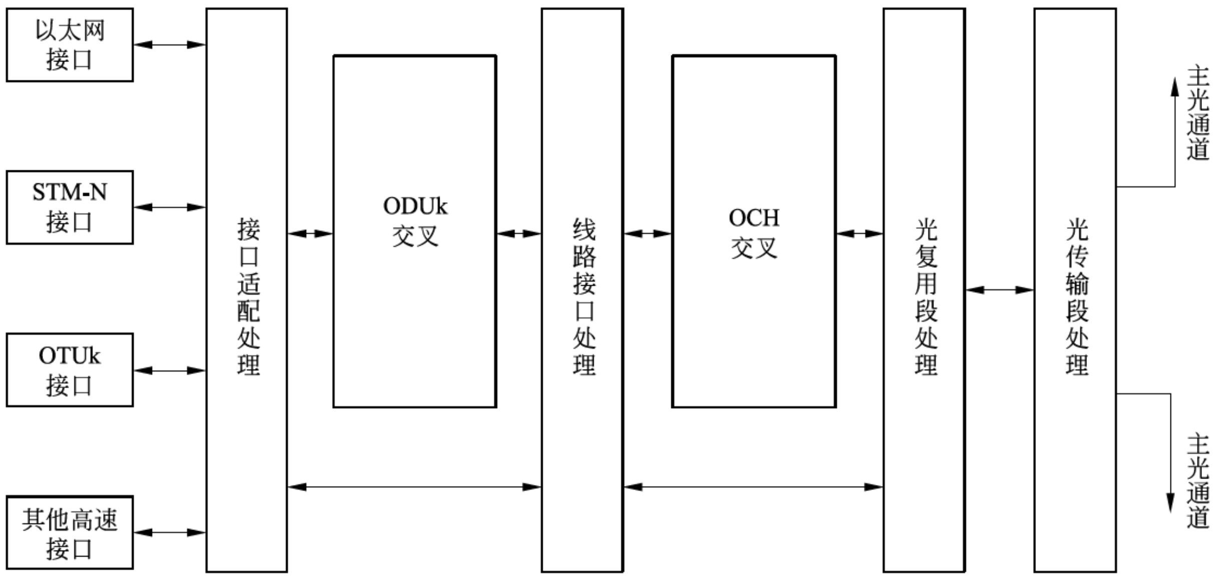


图 2　OTN 设备功能模型

参考文献

通信行业标准 YD/T 1990—2009. 光传送网(OTN)网络总体技术要求. 2009 （唐雄燕）

guang cunchuqi

光存储器(optical storage) 用光学方法从光存储媒体上读取和存储数据的一种设备。对光存储媒体最基本的要求是,存储单元的光学性质可以用物理方法改变以代表被存储的数据,同时这种性质改变可以用光的方法检测(读取)。目前几乎所有的光存储器都使用半导体激光器作为光源,因而光存储器也称为*激光存储器*。

光存储主要包括激光逐位存储、全息存储和其他光存储。广义来说,光存储器还应包括**条码阅读器**、光电阅读机等。但在计算机领域,光存储器一般指光碟机、全息存储器、光带机和光卡机等设备,其中光碟机应用最广。光碟机可分为**只读光碟驱动器**(如:CD-ROM,DVD-ROM 和 BD-ROM)及数字可重写光碟驱动器(如:CD ± R/RW,DVD ± R/RW,DVD-RAM,BD-R/RW,MO 和 PD)等多种类型。另外,全息存储系统也已实用化,产品有专用的全息存储系统(例如用于空间技术)和采用全息碟片作存储媒体的存储器。

光碟机主要由数据读写头、伺服机构和主轴系统、编解码和纠错系统、数据通道等部分组成,光碟机的主要技术指标是数据传输速率、寻道时间、可支持的最大容量、可支持的数据格式、记录的误码率和出错概率、可支持的接口标准。

光存储器技术发展很快,由 CD 驱动器发展到 DVD 驱动器和**蓝光光碟(BD)驱动器**。蓝光光碟(BD)已经成为光盘市场的主流产品之一,其单面单层容量可达为 25 GB,单面双层容量为 50 GB。

参考文献

1. 徐端颐,等. 高密度光盘数据存储. 北京:清华大学出版社,2004

2. Bradley A C. Optical storage for computers technology and applications. New York: John Wiley & Sons,1989

3. 张守仁. 光盘存储器. 北京:科学出版社,1989 （裴先登 黄浩）

guangdian jicheng dianlu

光电集成电路(optoelectronic integrated circuit, OEIC) 把光器件和电子器件做在同一基片上,完成光信息与电信息转换的一种集成电路。光电集成电路主要有两类:一类是完成光信息到电信息转换的电路,它由光电探测器、放大器及偏置电路组成,称为光接收器。常见的接收器件有光电晶体管、硅光电池等。另一类是完成电信息到光信息转换的电路,由光发射器件、驱动电路及偏置电路组成,称为光发射器。常见的发射器件有发光管、激光管、液晶。

光电集成电路根据其处理的光信息不同可分为红外光、可见光及激光三类。

光电集成电路已广泛用于照相机、电视摄像、工业自动控制、传真和光纤通信,以及机器人与视觉传感器、平面显示、夜视、卫星通信和导航等国民经济各个领域。计算机互联网络及电话光交换的研究,进一步扩大了它的应用范围,并促进光电集成电路从早期完成单路光电信息转换向多路及面阵方向发展。

为了实现图像的光电转换,在光电集成电路中必须集成调制器、多路复用及控制电路,从而使得现今的光电集成电路包含大规模的信息传输和处理电路。此外,还引进了各种新的器件以提高光电集成电路的性能价格比。例如,在红外与电视的摄像机中都采用了电荷耦合器件。

新器件的引进使光电集成的工艺比常规的集成电路工艺更为复杂化。传统的大规模集成电路是用平面工艺在硅基体上研制而成,其主要有源器件有双极晶体管、MOS 场效应晶体管及肖特基场效应晶体管三种,从而形成三类硅平面集成电路。但由于硅的禁带宽度(1.12 eV)只适合于制备可见光波段的探测器,而红外探测器必须在禁带宽度窄的化合物半导体上制备,激光器又必须在砷化镓衬底上制备,因此如何将三类器件基片统一起来是使光电集成电路从过去适合于光接收器的小规模光电转换电路,或适合于光发射器的小规模光电转换电路,发展到大规模光电集成电路所必须解决的关键问题。

为了使不同材料互补,利用异质外延工艺,在一种衬底材料上外延另一种衬底材料薄膜,发展出了有利于制备光电集成电路的复合衬底材料。例如,在硅片上异质外延砷化镓单晶薄膜,在衬底的硅面制作电子元件,在砷化镓薄膜上制作光子元件,就可以把基于硅的大规模集成电路技术与基于砷化镓的光子元件技术结合,提高集成度。除在硅面上异质外延砷化镓外,还可在砷化镓晶片上异质外延磷化铟单晶薄膜。

近年来发展出的硅基混合集成器件可以把光有源器件、光无源波导网络和电学布线都集成制作在一个硅基片上,以获得低成本、实用、多功能的器件。例如,将光发射器、光波导/调制器、光电探测器及驱动电路和接收器电路进行单片集成,构成硅基光电子集成回路。回路中所有器件均采用标准集成电路工艺制备,或是仅仅对工艺进行微小的修改,易于大规模生产。这种硅基单片光电子集成回路是一种电输入、光传输、电输出的互连系统,可实现光通信互连系统或集成电路芯片内的高速、高带宽信号传输,在光通信和高性能计算中有很大的应用潜力。

参考文献

陈弘达. 微电子与光电子集成技术. 北京: 电子工业出版社,2008

(林雨)

guangdie ku

光碟库(optical disc library)　一种以光碟为基本存储单元,可存放多片光碟,并由机械臂装置选取其中一片在光碟驱动器上装卸并进行存取数据的设备。又称**光盘库**。用于光碟库的光碟是普通光碟,所用的光碟驱动器也是普通的驱动器,不需要专用或特制的。只读光碟、一写多读光碟、可擦写光碟都可组成光碟库。

光碟库的基本结构包括: ①有许多插槽的光碟库体,用于插放光碟; ②一个或几个光碟驱动器; ③换碟机构,亦即将光碟从插槽中取出并装入驱动器或从驱动器取出并放回插槽的机械手; ④控制器,接受来自主机的命令并协调前三部分完成相应的动作。

光碟库内的碟片数一般为几十片到数百片,典型的驱动器数量为两个到六个。换碟时间(包括碟片运送时间、碟片装卸时间以及主轴达到工作转速的启动时间)的平均值约为 3 s 左右。采用的接口形式有 SCSI、SAS、SATA、USB 以及 LAN 等多种(参见**外存储设备接口**)。

与**磁盘阵列**等在线存储设备相比,光碟库的速度较慢,但信息适于长期保存,一般作为近线存储设备使用。

与**磁带库**相比,光碟库的容量较小,但数据保存的时间要长得多,随机检索的速度也比磁带库快,一直用于银行票据、保险资料、设计图纸、医疗图像、档案、专利文献等专门领域的信息归档保存。随着多层蓝光光碟等大容量光碟的出现,光碟库的容量有了数量级的提高,这就给光碟库在数据中心海量数据的长期保存领域找到了新的应用方向。大容量的光碟以及拥有数千甚至数万槽位数的大型光碟库将逐步进入过去大型磁带库占据的备份和归档市场,成为长期保存数据的主流技术。光碟库还具有绿色存储特性,在信息静态保存时不消耗能量,只有在读写时才耗能,这对于高耗能的数据中心而言是非常重要的特性。　(裴先登　谢长生)

guang jisuanji

光计算机(optical computer)　主要利用光技术和光器件而构造的计算机。同电子技术相比,光技术有如下显著的特点: ①传输速度快; ②互连数大,互连密度高,一般光学系统的互连数可达 106 个光点数; ③非物理触点互连,大大提高了可靠性和互连密度,同时,互连空间可以重复使用; ④遵循独立传输原理,上亿道光信息可以汇聚在一起,按照各自的目的地,独立、无干扰地传递信息; ⑤光的载波空间带宽可达 100 THz(电信号带宽最多为 100 GHz),而且信息传输无失真; ⑥功耗低,光信号衰减小。

光计算机可分为两大类: 模拟光计算机和数字光计算机。

1960 年第一台激光器的产生提供了空间和时间干涉性极好的光源,使得光学模拟计算机的研究走向了高潮。大量研究表明,模拟光计算机处理器借助相干光源和傅里叶变换特性可以获得实时处理大量数据的能力。但由于输入输出器件的限制,模拟光计算机处理器至今仍未能获得超过电子计算机处理器的性能。同时,所研究的光处理器主要处理一维和二维图像,其计算精度和适用领域均受到一定限制。

1979 年第一个半导体光学双稳器件研制成功。1983 年,提出了研究数字光计算机的构想。此后,全光数字计算机的研究进入高潮。虽然 1990 年宣布了第一台全光数字计算机研制成功,但由于工艺上的困难,光器件的性能、功耗、体积以及互连数等方面均不及电子器件,所以后来没有继续再向全光数字计算机方向发展,而倾向于采用光电混合的方法来研制光电混合型计算机。在这种计算机中,把硅和砷化镓集成技术用于实现传统的数字电路,而芯片、部件等器件之间的互连则由光器件来实现。发展这种光计算机的关键在于研制出体积小、能耗少、成本低且易于制造的光电子转换器,主要是微型激光二极管和光电二极管,同时要研制能精确控制

光路的制动定位系统。

参考文献

1. Murdoca M J. A digital design methodology for optical computing. Cambridge, MA: MIT Press, 1990

2. Ambs P. Optical computing: a 60-year adventure. Advances in Optical Technologies. Volume 2010, Article ID 372652, 2010

（刘德才）

guangxian chuanshu jiezhi

光纤传输介质（optical fiber transmission media） 用光信号传输信息所使用的介质。其优点在于：传输的速率高，传输质量高，传输距离远，保密性高，不受电磁噪声之干扰，体积小、重量轻、寿命长、价格低廉，绝缘、耐高压、耐高温、耐腐蚀，适于特殊环境之工作。

光缆（optical fiber cable） 由一根或多根光纤通过多层保护结构包覆而成的传输介质。

光纤（optical fiber） 一种利用光的全反射原理，在透明材料制成的纤维中传输光信号的介质。常见光纤主要有三层结构：核心层为高折射率石英玻璃芯，中间层为低折射率石英玻璃包层，最外是保护用的树脂保护层，如图1所示。光缆通信具有传输距离远、损耗低、频带宽、串扰小及抗电磁干扰能力强等特点。

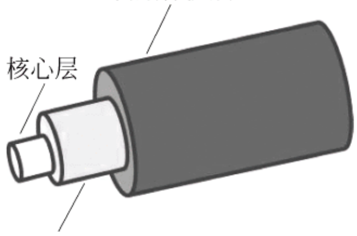


图1 光纤基本结构

根据光在光纤中的传输模式，光纤可分为单模光纤（single-mode fiber, SMF）和多模光纤（multi-mode fiber, MMF），其特性对比见表1。

表1 单模、多模光纤特性对比

单模	多模
给定工作波长内一个传播模式	给定工作波长内多个传播模式
长距离传输：100 km	短距离传输：2 km
模间色散小	模间色散大
核心层：9 μm、10 μm	核心层：50 μm、62.5 μm
常用波长：1310 nm、1550 nm	常用波长：850 nm、1310 nm
激光光源	LED 光源、激光光源

网络通信中通常用的光缆有以下几种规格：

（1）9 μm 芯/125 μm 外层，单模。

（2）10 μm 芯/125 μm 外层，单模。

（3）50 μm 芯/125 μm 外层，多模。

（4）62.5 μm 芯/125 μm 外层，多模。 （崔建）

guangxian yonghu huanlu

光纤用户环路（FTTx） 以光纤为传输媒体的接入网。该接入网中光纤是从网络干线到最终用户逐渐延伸的，网络运营商根据某种策略将光纤接到用户附近（或用户家中）的某个地点，在该处设置光网络单元 ONU，完成光电转换和分接等功能。FTTx 是指光纤在接入网中的推进程度和使用策略，并不是具体的接入技术。根据 ONU 的具体放置位置，光纤接入网可分为四种基本类型，即光纤到路边（fiber-to-the-curb, FTTC）、光纤到大楼（fiber to the building, FTTB）、光纤到办公室（fiber to the office, FTTO）、光纤到家（fiber to the home, FTTH）。

光纤用户环路的结构如下页图1所示。

（1）光纤到路边（fiber-to-the-curb, FTTC） 采用光纤作为传输介质并接到用户家路边的宽带接入体系结构。在 FTTC 结构中，数字信号从网络运营商经主干链路传到交换局，再从交换局传送到光网络单元（ONU）。在 ONU，光信号转换为电信号，然后通过铜线或同轴电缆传到用户，有时也可通过无线传输。电信运营商提供的服务使用双绞线从路边连接到客户场地。而电缆电视供应商则使用同轴电缆。FTTC 通常由使用多根光纤的交换网络实现，以传输双向信号流。FTTC 也可同 HFC（hybrid fiber-coaxial 的缩写，即混合光纤同轴电缆网）体系结构配合使用（参见**混合光纤同轴电缆**）。FTTC 网络从 ONU 到每个用户的下行信息的传送速率为 51 Mb/s，上行速率则为 1.62 Mb/s。以 51 Mb/s 的传输速率交换信息流可同时携带 6～7 个高质量的数据流到每个用户。FTTC 能为每个用户提供各种高速数据服务和电话服务，并可为每个用户提供相当可观的

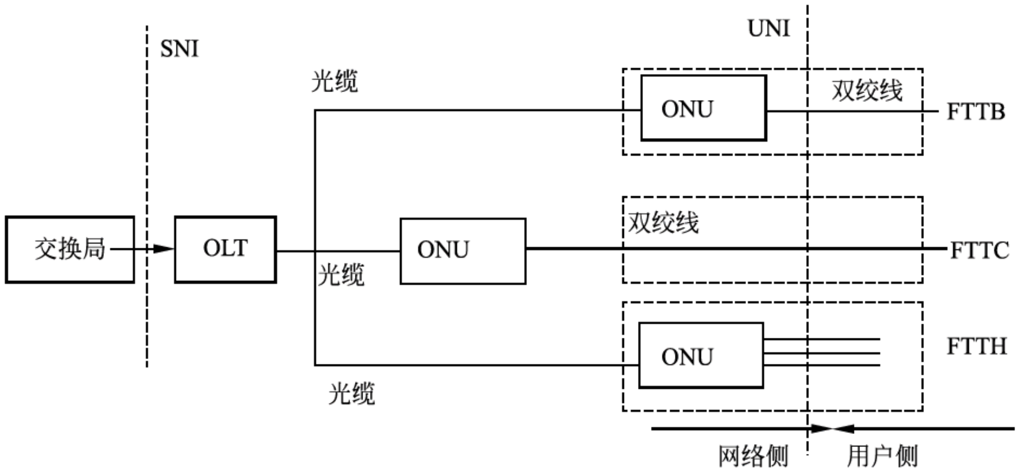


图 1　光纤用户环路

OLT：光线路终端(optical line terminal)，ONU：光网络单元(optical network unit)

频带。然而，对 FTTH 这类网络结构目前还没有很大的客户需求，因为目前使用 VDSL 双绞线结构已能有效地、廉价地提供能满足用户需要的频带。在 FTTC 结构中，现有的铜缆资源仍能利用，具有较好的经济性。FTTC 预先建设了一条靠近用户的宽带传输链路，一旦有宽带业务需求，就可以很好地将光纤引至用户处。由于光纤已经非常靠近用户，因此可以充分发挥光纤化的优越性。综合考虑初次投资和维护运行费用，在提供 2 Mb/s 以下窄带业务时，该方式目前是有源光接入网最为经济的实现方式。

(2) 光纤到楼(fiber-to-the-building, FTTB) 是 FTTC 的变型，不同之处是 ONU 直接放在楼内，再经铜线将业务分送到各个用户。FTTB 是一种点到多点的结构，即一个 ONU 为多个用户提供接入。FTTB 比 FTTC 的光纤化程度更进一步，因而适合高密度用户区。由于 ONU 靠用户更近，铜缆更短，因此更容易满足用户的高带宽接入需求。

(3) 光纤到办公室(fiber-to-the-office, FTTO) 将 FTTC 结构中的 ONU 放置在办公室，即为 FTTO。FTTO 实现了全程光纤接入，主要用于大型企事业单位，业务量需求大，一般采用环型或点到点的结构。

(4) 光纤到家(fiber-to-the-home, FTTH) 在 FTTC 结构中 ONU 直接放置在用户家中，同 FTTO 一样是全程光纤接入。不同的是 FTTH 用于家庭，从业务量和经济性考虑一般采用点到多点结构。

FTTO 和 FTTH 都无任何有源设备，是一个能够提供宽带接入的透明网络，是用户接入网的长远目标。

参考文献

1. 胡道元. 智能建筑计算机网络工程. 北京：清华大学出版社，2002

2. Warrier P, Kumar B. XDSL architecture. New York：McGraw-Hill Inc.，2000

3. 柯赓. 接入网技术与应用. 西安：西安电子科技大学出版社，2009

4. 雷维礼，马立香，等. 接入网技术. 北京：清华大学出版社，2006

(程时端　马严)

guangxian genzong jishu

光线跟踪技术(ray tracing technique)　通过模拟光线在环境中的走向轨迹进行图形绘制的一种全局光照技术(参见**光照模型**)。光线跟踪技术于 20 世纪 60 年代末提出，用于确定阴影和可见面。70 年代末 80 年代初将该技术应用到**真实感图形生成**的复杂光照计算，包括反射、透射等。该技术的发展和应用显著地改善了计算机生成的真实感图形的质量，因而光线跟踪成为**计算机图形学**中具有重要地位的技术。

由环境中的物体产生真实感图形的过程是一个从三维空间向二维图形屏幕投射的过程。屏幕上每一像素所具有的色彩和亮度应精确地反映整个环境中的复杂光照在该点投影方向上所产生的综合效果。光线跟踪技术正是从这一原理出发，以观察点为起始点，通过屏幕上每一像素点沿着光线投射的逆方向去“追根溯源”，累积各种光照能量。

光线跟踪有时也可称为光线投射。但光线投射现在通常是指从视点出发，经像素向物体空间进行

一次性投射。这种投射方法在该技术发展初期曾用于探测可见面。而光线跟踪一般是指可处理反射、折射等光照效果的复杂多层的跟踪技术。如图 1 所示,跟踪光线从视点出发通过屏幕交于物体 A 点,其光亮度 I 由 3 部分光强分量组成:光源直接照射,即由局部光照模型计算的光亮度(L_1),物体表面反射光线方向上的光亮度(R_1)以及透射方向上的光亮度(T_1)。继续沿反射方向和透射方向跟踪,又有

$$R_1 = R_{11} + T_{11} + L_2, \quad T_1 = R_{12} + T_{12} + L_3$$

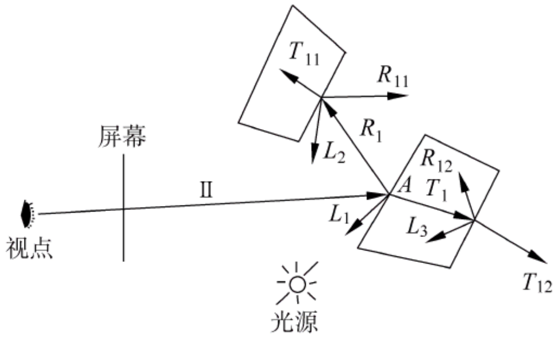


图 1　光线的递归跟踪

如此继续跟踪下去,其递归跟踪过程便形成了一棵光线跟踪树。该树的层次深度根据需要而定。树的每一个结点上的光亮度为该结点子树的综合光亮度。实际计算时是从树的最底层即叶结点的光亮度算起,沿着光线跟踪的反方向逐层归纳各子树的光照,从而得到树根处的值,即为该像素处最终的光照效果。

由于使用光线跟踪方法很容易沿着光线求取反射及透射效果,因而该技术不仅有利于显示阴影,而且很适合于模拟高光反射即镜面反射、透视等光照效果。

参考文献

1. 唐荣锡,汪嘉业,彭群生,汪国昭,等. 计算机图形学教程(修订版). 北京:科学出版社,2000

2. 孙家广,等. 计算机图形学. 3 版. 北京:清华大学出版社,1998

(吴恩华)

guangzhao moxing

光照模型(illumination model)　用于计算物体表面光照强度的数学模型。景物表面的颜色和明暗色调是表现景物的几何形状、表面材料属性以及所处环境的最重要手段。在**计算机图形学**中,用物理学中的光照强度表征物体辐射光能的强弱和色彩组成。为了生成真实感图形,必须根据光学原理计算物体表面各处的色彩和光照。当物体受到光源照射时,光线可以通过物体表面反射产生反射光;如果物体是透明体,光线还可穿透物体产生透射光。反射光和透射光进入人眼产生视觉效果。

光照模型通常可分为局部光照模型和全局光照模型两大类。局部光照模型只考虑光源直接照射所引起的反射光强,模型简单,计算量较小;全局或整体光照模型则全面考虑环境中物体之间光能的多重反射、透射以及相互影响。**光线跟踪技术**、**辐射度技术**和**光子映射**方法是全局光照模型最重要的方法。

就局部光照而言,对特定光源在物体表面每一点产生的光照强度进行光亮度计算,可以仅考虑简单实用的光照模型,物体不透明,无透射光线,因而物体表面呈现的颜色和亮度仅由反射光决定。反射光通常由环境光、漫反射和镜面反射 3 个分量组合而成,并可用下式表示 $I = \mathrm{ka} * \mathrm{Ia} + \sum(\mathrm{kd} * \mathrm{Id}i + \mathrm{ks} * \mathrm{Is}i)$,式中,Ia,Id$i$,Is$i$ 分别表示环境光、漫反射和镜面反射的光照强,ka、kd、ks 分别为环境光反射、漫反射和镜面反射的比例系数,这些系数取决于物体表面的属性。其中后面两项分别要对每个光源(i)进行累加计算。环境光反射分量是由物体周围的环境光产生的。假定光线均匀地从周围环境入射至景物表面,并等量地向各个方向反射出去,因而环境反射光分量为常量。漫反射分量是光源入射后经物体的粗糙表面向各个方向均匀反射而引起的,其大小与接受方向无关。漫反射计算服从 Lambert 定律:反射光强与光源入射角的余弦成正比,即 $\mathrm{Id}i = \mathrm{I}i * \cos(\alpha i)$ 式中,Ii 为光源 i 的光强,αi 为该光源的入射角(是入射方向 L 与法线方向 N 的夹角);镜面反射是由物体表面光滑特性引起的带有方向性的反射。它遵守光的反射定律:反射光线与入射光线相对于表面法线对称。使用时,镜面反射分量常使用以下公式来模拟:$\mathrm{Is}i = \mathrm{I}_i * (\cos^n(\beta_i))$ 式中,β_i 为反射光线向量 $\boldsymbol{R}$ 与视线向量 $\boldsymbol{V}$ 之间的夹角,n 为正整数(通常取 $2 \leqslant n \leqslant 200$)。这样,一个简单实用的光照计算模型可表示为 $I = \mathrm{ka} * \mathrm{Ia} + \sum i[(\mathrm{kd} * \cos(\alpha_i) + \mathrm{ks} * \cos^n(\beta_i)) * I_i/(d_i * di + C)]$,式中,ka,kd,ks 分别为环境光反射、漫反射和镜面反射的比例系数,这些系数取决于物体表面的属性;di 为光源至物体的距离;C 为一特定常数。以上计算模型于 1973 年由 B. T. Phong 提出,因而称为 Phong 模型。

在应用问题中,曲面物体常常近似地表示为一组多边形小平面片,称为多边形网格。对于整个曲面进行局部光照计算,即明暗处理(shading)的计

算，通常是基于多边形网格的法向量采用扫描线方法对这些小平面片进行插值计算，以达到快速计算的目的。有两种不同的插值算法，分别称为 Phong 明暗处理方法和 Gouraud 明暗处理方法。Phong 明暗处理方法(Phong shading)在对每个多边形进行计算处理时对面片的法向量进行线性插值,从而在每一像素点上根据插值所得的法向量按光照模型计算光强(如利用以上 Phong 模型进行光亮度计算)。算法的具体过程是：首先计算多边形顶点处的法向量,然后应用双线性插值求得多边形内扫描线上每个像素处的法向量,再根据法向量计算光强。Gouraud 明暗处理方法(Gouraud shading)是沿扫描线对光强作线性插值,而不是如 Phong 明暗处理方法那样对法向量进行线性插值。该算法的具体处理过程类似于 Phong 明暗处理方法：首先在多边形顶点处求法向量(其法向量通常用围绕顶点周围的多边形法向量的平均值来表示),然后使用局部光照计算模型计算顶点处光强。多边形内扫描线上各像素点处的光强直接由多边形顶点处的光强经双线性插值得到。

Gouraud 明暗处理方法直接对光强值而不是对法向量进行插值,因而它比 Phong 方法计算量小,简单实用。为此,该方法得到更广泛的应用。在许多高性能的图形工作站中,该方法作为图形加速器或基本图形软件业已实现的功能,可提供给用户直接使用。因为 Phong 明暗处理方法是对向量进行插值,比 Gouraud 明暗处理方法计算量大,但有时在局部范围内对曲面表面光照的模拟更精确,效果更好。

参考文献

1. 唐荣锡,汪嘉业,彭群生,汪国昭,等. 计算机图形学教程. 修订版. 北京：科学出版社, 2000

2. 唐泽圣,周嘉玉,李新友. 计算机图形学基础. 北京：清华大学出版社,1995

3. Foley J D, Dam A V. 交互式计算机图形学基础. 唐泽圣,周嘉玉,等译. 北京：清华大学出版社,1986

4. Cohen M F, Wallace J R, Hanrahan P, Greenberg D P. Radiosity and realistic image synthesis. San Francisco, CA：Morgan Kaufmann, 1993 （吴恩华）

guangzi yingshe

光子映射(photon mapping) 一种基于光线跟踪的全局光照明方法(参见**光照模型**),包含两个主要步骤：首先,大量离散的光子从光源出发,经过若干次折射和反射分布到场景之中;然后,从摄像机投射出众多视线与场景相交,根据交点周围光子的分布来估算最终像素的颜色。

第一步是模拟光源向外发射光子和光子在场景中的传播过程,生成场景中的光子分布图,称为光子图(photon map)。光源随机地向外发射光子,所有光子具有相同的能量,发射方向与光源的能量辐射分布函数一致。光子在场景中沿直线传播,当它到达景物表面时,根据其物体表面的材质属性可能被反射或吸收。除非景物表面是纯镜面的,否则光子的位置、能量和方向将被存储于光子图中。一个光子在传播路径上可能被存储多次,但是存有不同的位置、能量和方向。

第二步是基于光线跟踪技术对场景进行绘制,计算每一个像素所接收的辐射能。根据绘制方程,对场景中任意一点的辐射能可以分解为三部分：直接光照、镜面反射以及间接光照。其中直接光照和镜面反射部分可以用**光线跟踪技术**很好地处理,而间接光照部分则采用光子图进行有效地计算。其基本方法是：首先在光子图中查找距该点最近的一些光子,然后把这些光子作为间接光源,计算和累加它们对辐射能的贡献。为了提高光子的查找速度,光子图一般以平衡 KD 树形式存储。

当场景中存在透明物体或半透明介质时,光子映射方法通过在光子跟踪过程中考虑光子折射和散射现象,模拟焦散和半透明介质的效果。为了减少光子数量,提高绘制效率,折射和散射的光子被存储为两个独立的光子图,分别称为焦散光子图(caustics photon map)和体光子图(volume photon map)。

与辐射度方法相比,光子图和场景表示是完全分离的,而且光子映射不需要网格剖分,所以可以有效地处理大规模场景。与双向光路跟踪、蒙特卡罗光线跟踪等方法相比,尽管光子映射需要额外的计算并存储光子图,但是其效率高,并且可以处理所有的全局光照明效果。需要注意的是,光子映射是一种是有偏差的方法,其绘制误差无法通过多次绘制取均值的方法消除。减少绘制误差的一个方法是增加光子的数量。

参考文献

1. Henrik Wann Jensen. Global illumination using photon maps. In：Rendering Techniques' 96, 1996：21-30

2. Henrik Wann Jensen. Realistic Image Synthesis Using Photon Mapping. Natick, MA：A K Peters,

2001　　（刘新国　戴强）

guangpu yuyan

广谱语言（wide spectrum language）　可在不同的抽象级别上书写**软件需求定义**、**功能规约**、**设计规约**及实现的语言。

好的软件规约要在较高的抽象级上说明做什么，强调明晰，而好的实现必须针对虚拟机给出如何做的详细描述，强调功效。明晰和功效很难在一个程序中同时得到满足，这两者之间有很大距离。程序（规约）在这个距离间分成若干个抽象级别。广谱语言及其相应的支撑系统能够在同一个语言架构下从明晰的规约通过推理逐步向高效的程序转化，或者提供低级程序相对于高级规约的正确性证明的辅助手段。

广谱语言应提供规约级到实现级的数据结构和控制结构的构造，并提供结构间语义等价的转换规则，用于自动转换或验证低级结构相对高级结构的正确性。其代表性的语言是 CIP- L。　　（伊波）

guidang cunchu

归档存储（archival storage）　指用以保证数据集合的一致性复制的技术。通常用以长期持久地保存事务或者应用状态的记录。一般情况下，归档存储通常用于审计和分析的目的，而不是用于应用恢复的目的。

归档存储所考虑的首要问题是数据的永久保存，不管时间多久、技术如何改变，归档数据必须保证真实性、可读性和可理解：

（1）真实性　数据内容和有关背景信息与原始数据一致，并不可篡改；

（2）可读性　数据本身及相关信息可读可用；

（3）可理解　数据的格式、内容和上下文关系可理解。

为保证归档数据的真实性，较为普遍采取的解决方案是采用 WORM（write once，read many，一次写，多次读）技术。即通过硬件设备的控制使存储介质只能写入一次数据，而不能重复写入且不允许修改，因此，可保证数据安全及其完整性。

光存储具有物理 WORM 功能，在归档存储方面占有重要地位。但毕竟容量、速率有限，因此出现了具有 WORM 功能的磁盘存储系统和**磁带库**。

基于内容寻址存储（content-addressable storage，CAS）技术实现了支持 WORM 的磁盘归档系统。在内容寻址存储中，获取数据不是按照用户定义的文件名和位置，而是按照数据内容的数字指纹寻址。数字指纹是根据文件内容通过某种特定算法计算出来的字符串，常用的方法是根据数据内容计算出固定长度的哈希（Hash）。数字指纹是数据对象的唯一身份标识号码（identity，ID），可以用于 WORM 和内容认证，CAS 可以自动检测数据对象的变化，实时保护数据对象不被恶意修改，维持数据对象的完整性。

基于磁盘的内容寻址存储（CAS）归档方案通过采用新的技术，如 SATA（串行 ATA 接口）磁盘组成的**磁盘阵列**和虚拟磁带库、**分级存储**管理和**存储虚拟化**技术，可提高归档服务水平，但成本较高。具有 WORM 功能的磁带库是目前成本效益最好的解决方案，不仅在于每 GB 的分摊成本上，而且磁带库系统功耗也比磁盘系统低得多。

归档存储还需要考虑提供对数据的方便访问。通过对数据采用分类的方式组织、基于对象进行描述、基于元数据对数据进行搜索来提高可访问性，几十年后对数据的读取可以像现在对本地或分布系统的读取一样方便。

归档存储可应用于以文件为主的非结构化数据、以邮件为主的半结构化数据和以数据库为主的全结构化数据。应用最普及的是文件归档，主要针对电子文档、图片、扫描和数码照片、音视频文件等非结构化数据。

随着**云存储**技术的发展，将数据迁移到云存储中长期保存的云归档成为一个新的选择。云归档解决方案都是付费即用型的，对于规模小的企业也是一个好的选择。企业实施归档存储的价值主要体现在法规遵从、存储优化、数据利用等方面。从法规遵从来说，企业很多的信息，如合同、设计图纸、技术方案、项目文档等的存储必须要遵从相应的法规，具有 WORM 功能的归档存储系统将满足企业在法规遵从方面的需要；随着数据量的增加，将存储中不改变的数据存放到归档系统中，可以显著减少存储系统的负担，并采用重复数据删除等高级技术减少冗余数据，以实现存储优化；归档的数据通过良好的数据组织能力，提供高效数据访问与搜索能力，如通过查看历年合同作为现在项目的参考依据等，可将数据的价值发挥到最大。

参考文献

1. You L，Karamanolis C. Evaluation of efficient

archival storage techniques. Proceedings of the 21st IEEE/12th NASA Goddard Conference on Mass Storage Systems and Technologies,2004

2. Peterson M, Zasman G, Mojica P, et al. 100 year archive requirements survey. Storage Networking Industry Association. http://www.snia.org/sites/default/files2/100YrATF_Archive-Requirements-Survey_20070619.pdf, 2007

3. Zhu Li-Gu,Sun Zhi-Wei,Liu Hao. Distributed archiving storage system based on CAS. 2009 IEEE International Conference on Virtual Environments, Human-Computer Interfaces, and Measurements Systems (VECIMS 2009) Proceedings. Hong Kong,China, May 11-13 2009, 365-368 (朱立谷　谭玉娟)

guijie fangfa

归结方法(resolution method)　**一阶逻辑**中的一个完备的推理方法。亦即对于一阶逻辑中任一个恒真公式,使用此方法都能在有限步内证明这个公式的恒真性,因此归结方法成为**机器证明定理**的一种重要方法。

从17世纪数理逻辑诞生起,人们就期望得到一种机械的逻辑方法,用这个方法能够证明用数理逻辑描述的所有定理。在长期的研究中,虽然得到了一些算法,但是这些算法在计算机上实现时,都因为过于复杂而不能真正地完成证明定理的任务。直到1965年John Alan Robinson提出归结方法,才使得在计算机上证明定理向前跨进一大步。其后,各国学者对这一方法进行了一系列改进。1965年,Wos,Robinson和Curson等人提出了支持集归结,支持集归结是完备的。1967年,Slagle提出了语义归结。1970年,Loveland和Luckham提出了线性归结。1971年,Boyer提出了锁归结。锁归结是完备的。1979年,刘叙华将锁归结与语义归结相结合,提出了锁语义归结。1982年,王湘浩和刘叙华提出广义归结。

下面以**命题逻辑**为例介绍这一方法。我们将原子或者原子的否定统称为文字。由有限个文字组成的析取式称为子句。例如,$A \vee B \vee \neg C$是子句,其中A,B,C是原子。$(A \wedge B) \vee C$不是子句。

归结式:设C_1和C_2是两个子句。若在C_1和C_2中分别存在两个文字L_1和L_2,并且$L_1 = \neg L_2$,不妨设

$$C_1 = L_1 \vee C'_1$$

$$C_2 = L_2 \vee C'_2$$

其中C'_1和C'_2也都是子句,则$C'_1 \vee C'_2$称为对C_1和C_2使用归结方法得到的归结式。可见,归结方法是一种操作起来非常简洁的方法:在两个子句中删掉互补的两个文字(如果存在的话),将剩下的子句再析取起来。例如,$C_1 = \neg A \vee B$,$C_2 = A$,则对C_1和C_2使用归结方法,得到B。而C_1也可等价地写成$(A \rightarrow B)$,C_2是A,由C_1和C_2得到B的推理实际上是三段论推理。因此,三段论是归结方法的一个特例。

用归结方法证明数学定理实质上是一个反驳过程。在数理逻辑中,一个数学定理,通常写成如下公式:

$$A_1 \wedge A_2 \wedge \cdots \wedge A_n \rightarrow B$$

其中$A_1,\cdots,A_n$是前提公式,B是结论公式。证明上述定理成立,相当于证明公式$(\neg A_1 \vee \cdots \vee \neg A_n \vee B)$是恒真的,也相当于证明如下公式:

$$\neg(\neg A_1 \vee \cdots \vee \neg A_n \vee B) = A_1 \wedge \cdots \wedge A_n \wedge \neg B \qquad (*)$$

是恒假的。将公式等价地化成合取范式,得到一个子句集S(S中子句之间是合取关系),用归结方法证明了S的恒假性,就相当于证明了上述定理。例如,要证明如下定理:

$$((A \rightarrow B) \wedge (B \rightarrow C) \wedge A) \rightarrow C$$

只要证明公式$((A \rightarrow B) \wedge (B \rightarrow C) \wedge A \wedge \neg C)$为恒假即可。将此公式化成合取范式:

$$(\neg A \vee B) \wedge (\neg B \vee C) \wedge A \wedge \neg C$$

将上面公式写成如下子句集S:

$$S\begin{cases}(1) = \neg A \vee B \\ (2) = \neg B \vee C \\ (3) = A \\ (4) = \neg C\end{cases}$$

(1)和(3)作归结得到B,B和(2)作归结得到C,C和(4)是一对矛盾,作归结得一恒假子句(称为空子句),因此S是恒假的,从而证明了定理。

如此简洁的方法却有着很强的功能。

完备性定理:子句集S是恒假的,当且仅当从S出发,使用归结方法可推出一对矛盾(即得到一个空子句)。

这一方法不仅对命题逻辑适用,对一阶逻辑也适用,当然需要加进合一替换的概念。在20世纪90年代先后出现了基于归结方法的高效一阶逻辑定理证明器Otter和Vampire。

归结方法也被引进各种非经典逻辑中,成为**自动推理**领域中的一种基本方法。

参考文献

1. 刘叙华. 基于归结方法的自动推理. 北京：科学出版社，1994

2. Leitsch A. The resolution calculus. Birlin：Springer，1997

（刘叙华　欧阳丹彤）

guina tuili

归纳推理（inductive inference）　一种由特殊到一般、由现象到本质的推理方式。归纳推理通常按获取数据、整理加工数据、分析数据、得出结论等步骤，根据反映客观事物的事实以及各种表象信息，由表及里、由浅入深、由此及彼地发现客观事物的内在性质、发掘相互间的联系或者得出可用于解释普遍现象的一般性规律，形成理性认知。

从前提所考察对象的范围来看，归纳推理可以分为完全归纳推理和不完全归纳推理两类。完全归纳推理是完整地考察客观事物所涵盖的每个实际对象，准确地得出关于该类事物的一般性结论，数学归纳法就是一种完全归纳推理方法；不完全归纳推理则仅考察客观事物涵盖范围中的一部分对象，分析后得出结论，预测该结论也适用于其他对象，由于推理中前提的覆盖范围小于结论的覆盖范围，因此，其结论具有或然性。

从推理方法来看，经典的归纳推理方法包括“简单枚举法”“科学归纳法”“典型归纳法”“溯因法”“穆勒五法”等，现代归纳方法有“统计归纳法”“概率归纳法”等。

简单枚举法：

$s_1, s_2, \cdots, s_n$ 都具有属性 P

$s_1, s_2, \cdots, s_n$ 都是 S 类中的部分对象，且尚未遇到反例

S 类中的对象都具有属性 P

简单枚举法能够直接从观测对象联系到本质性结论，是最简单、基本的归纳推理方法。虽然其容易发生“以偏概全”的逻辑错误，但扩大前提对象的考察范围，能够有效提高其可靠性。如果在简单枚举归纳前提中考察所有可能对象，那么它就成为完全归纳推理。

科学归纳法：

$s_1, s_2, \cdots, s_n$ 都具有属性 P

$s_1, s_2, \cdots, s_n$ 都是 S 类中的部分对象，且 S 与 P 有某种必然联系

S 类中的对象都具有属性 P

虽然科学归纳法和简单枚举法都是不完全归纳推理，但两者间有明显差别。科学归纳法是建立在对前提中的事物与属性间因果联系的科学分析基础之上的，因而其结论的可靠程度取决于前提中事物与属性之间因果相关程度的强弱，而与事例数量无关。

典型归纳法：

s_1 具有属性 P

s_1 是 S 类中具有代表性的对象

S 类中的对象都具有属性 P

典型归纳法同样不关心前提中所考察对象的数量，只要具有典型意义，那么一个对象就可以归纳出结论。它虽然具有迅速有效完成对前提考察的优点，但由于依赖于推理主体的经验知识来判断对象是否具有典型性，所以也有陷入“以偏概全”错误的可能。

溯因法：

若 P 则 Q

已知 Q 成立

则 P 可能成立

溯因法由结果追溯原因，与三段论演绎推理过程恰好互逆，在科学研究和社会实践中有着广泛应用。

归纳推理方法灵活多样，往往与具体研究问题相关，具有很强的应用性。所得结论通常具有很强的或然性，需要通过实验验证其可靠性或演绎证明其正确性。相对于演绎推理，归纳推理还难以建立起形式化的公理系统，具有非形式化的特点。

归纳推理能够由个别推广到一般，将有限扩展到无限，从局部扩张到全局，起到连接外在事实现象与内在本质规律的桥梁作用，是科学发现的一种重要手段，与演绎推理相辅相成、紧密相关，共同构成思维的基本方式。随着人工智能应用对知识获取技术越来越迫切的要求，以及以演绎推理为基础的**自动推理**技术逐步成熟，人们开始更多地将注意力转向包括归纳推理在内的各种机器学习技术和方法的研究。

参考文献

1. 张志成. 逻辑学教程. 2 版. 北京：中国人民大学出版社，2006

2. 中国人民大学哲学系逻辑教研室. 形式逻辑. 北京：中国人民大学出版社，1984

3. Brozdil P B. Machine learning：ECML-93. Berlin：Spring-Verlag，1993

（赵沁平）

guiyue jisuanji

归约计算机（reduction computer）　一种面向

函数式程序设计语言的**非传统计算机**。

冯·诺依曼计算机所用的命令式程序设计语言(例如 FORTRAN,PASCAL 等)难以编写、验证和重用程序,不适合并行处理,因而有人提出函数式程序设计语言(参见**函数式程序设计语言**)。这种语言有利于保证程序各部分的并行执行,更适合于编写高度并行系统结构的并行处理程序。

归约机的处理对象是表达式,其处理过程是表达式的归约过程。归约的计算模型是把函数式程序看作嵌套的表达式,它的执行过程就是求出表达式的值来代替表达式本身。表达式的求值,是由一系列表达式归约组成,而表达式一次归约是一次函数应用,或是一个子表达式的变换。例如,内积的定义为:

IP=(+(AA,*),TR),求数组(1,2,3,4)和(5,6,7,8)的内积的归约过程如下:

(1)〈IP,((1,2,3,4),(5,6,7,8))〉

(2)⇒〈(+,(AA,*),TR),((1,2,3,4),(5,6,7,8))〉

(3)⇒〈+,〈(AA,*),〈TR,((1,2,3,4),(5,6,7,8))〉〉〉

(4)⇒〈+,〈(AA,*),((1,5),(2,6),(3,7),(4,8))〉〉

(5)⇒〈+,(〈*,(1,5)〉,〈*,(2,6)〉,〈*,(3,7)〉,〈*,(4,8)〉)〉

(6)⇒〈+,(5,12,21,32)〉

(7)⇒70

一个操作符及其变元的组合称为可归约表达式。在嵌套的多个表达式中已计算出其变元值的可归约表达式称为最内层可归约表达式。归约过程就是将每个最内层可归约表达式用其值来替换,这样又形成了新的最内层可归约表达式,然后再对其进行归约。最后,整个程序全部被归约,仅留下最终结果。

从系统结构看,归约机是一种以需求驱动方式进行操作的计算机。在它的控制图中,一个节点上的操作仅在需要用到它的输出结果时才开始启动,如果此时这个节点的输入数据还未能获得,就由这个节点再去启动能获得它的各个输入数据的节点,就这样把需求链一直延伸下去,直到遇到常数为止。然后再按反方向逐级进行操作,把输出结果返回给需要它的上一级节点,直到最初发出需求的节点为止。需求驱动只计算那些在获得最终结果时需要其输出的节点,它执行的是最低限度的计算工作,它比数据流计算机工作效率高,且节省资源。

归约可分为串归约和图归约两类。在串归约机中,表达式、函数定义和目标以字符串形式存储,函数式程序可以不经翻译直接输入串归约机中处理。在图归约机中,表达式、函数定义和目标以图的形式存储,处理的对象是图。串归约总是优先处理最内层可归约表达式,当多个操作要求使用同一个数值时,计算该值的公共子表达式要为每个操作分别求值,形成该值的多份副本送往不同的操作,这种“按值调用”机制会使系统的使用效率降低。图归约实行“按引用调用”机制,一旦计算出一个表达式的值,就保存起来供再次使用,其他操作要求使用该值时,不必重新求值,这样可以减少重复计算工作量,但要能访问已计算好的共享数据,就要添加指针和存储地址,增加复杂性。

归约机是在20世纪70年代末开始研究的,但仅对其系统结构做过一些试验性的工作。在有影响的项目中,串归约机有美国北卡罗来纳州立大学的细胞树机等,图归约机有 ALICE 系统、GRIP 系统和 Flagship 系统等。 (孙强南 郑纬民)

guifanhua

规范化(normalization) 对数据模式中的属性进行有效组织的过程。

规范化是将非规范模式或规范化程度低的关系模式根据其中数据之间的关系分解为两个或多个规范模式的过程。规范化的主要目的是消除冗余数据和确保数据的依赖性处于有效状态(存在依赖关系的数据存储在一个表里)。关系模式中存在的冗余数据是进行规范化的主要原因,冗余数据可以造成冗余存储、更新异常、插入异常、删除异常等问题。规范化过程能够减少数据库和表的空间消耗,并确保存储的一致性和逻辑性。

规范化的方法是把泛化的关系模式分解成单一化的关系模式,把泛化的模式中对非键码属性集的依赖变成单一化的模式中对键码的依赖,从而消除关系模式内的不合适的数据依赖,达到简化数据库修改和减少数据冗余的目的。例如关系模式 R(工号,工资级别,工资额)中的工号是键码,工资额函数依赖于工资级别而后者不是键码,所以不是第三范式的关系模式。按此模式构造的数据库中,工资级别和工资额的对应关系有大量重复(有许多职工工资级别相同因而工资额也相同)且不完整(有的级别没有职工因而也没有工资额)。若改成 R_1(工

号,工资级别),R_2(工资级别,工资额),则两者都是第三范式,用此模式构造的数据库中消除了数据冗余,也完整地反映了工资级别和工资额的对应关系。一般情况下这种分解不止一种方式,重要的是分解过程中不能丢失信息,即根据分解后的关系能还原成分解前的关系。

参考文献

1. Ullman J D. Principles of database systems. 2nd ed. London: Pitman Publishing Ltd., 1987

2. 施伯乐,丁宝康,汪卫. 数据库系统导论. 3版. 北京:高等教育出版社,2010 (王鹏　施伯乐)

guoji dianxin lianmeng

国际电信联盟(International Telecommunication Union,ITU)　世界上历史最悠久的信息通信技术的标准化组织之一,也是最具有影响力的国际网络标准化组织之一。作为联合国的下属机构,ITU具有较强的政府背景,在协调和制定全球范围内的技术标准方面发挥了重要的作用。

国际电信联盟的前身是1865年成立的国际电报联盟(英文缩写也是ITU),它在1924年成立了国际电话咨询委员会(CCIF),1925年成立了国际电报咨询委员会(CCIT),1927年成立了国际无线电咨询委员会(CCIR),1934年更名为国际电信联盟。1947年ITU正式成为联合国(UN)的下属机构。1956年,CCIF和CCIT合并成为国际电话电报咨询委员会(CCITT)。为推动改善发展中地区的通信发展,ITU在1989年成立了电信发展局(TDB)。1992年ITU进行了大规模的组织架构的整合,形成目前的组织架构:电信标准化部(ITU-T),无线电通信部(ITU-R)和电信发展部(ITU-D)。

ITU-T主要负责协调全球电信方面的技术、政策研究和标准制定;ITU-R主要负责协调全球无线电通信方面的技术、政策研究和标准制定;ITU-D主要负责全球发展中地区的电信技术、政策研究和标准制定。与网络相关标准主要由ITU-T负责制定,目前关注信息与通信技术(ICT)方面标准的制定。

ITU-T制定的标准称为ITU-T建议,要求全体成员同意才能生效,4年修订一次。ITU-T主要以研究组(SG)形式开展相关的网络国际标准的研究工作。由于ITU-T在每个研究周期都将更新研究组设置,现有的研究组编号并不连续。

ITU-T的工作方式有几种。采用**全球标准化行动**(GSI)的组织形式,研究涉及多个研究组的技术标准,例如NGN-GSI(下一代网全球标准化行动)和IoT-GSI(物联网全球标准化行动)等。采用焦点组(FG)的组织形式,启动可以由非ITU成员参与的国际标准化研究,例如FG M2M(机器到机器业务层焦点组)。采用**联合协调活动**(JCA)的组织形式,协调ITU-T与其他国际标准化组织之间的标准化研究和制定工作,例如JCA-IoT(物联网联合协调活动)主要协调ITU-T与ISO等其他国际标准化组织之间有关物联网的标准研究和制定工作。

ITU-T TSAG(电信标准咨询组)作为ITU-T的研究组、成员和职员的咨询机构,负责ITU-T标准化工作的管理和标准的审批。目前已经制定了新项目立项和国际标准的流水线方式的审批流程,将国际标准审批时间从原来的4年时间压缩到了现在的8周。

参考文献

http://www.itu.int/en/pages/default.aspx

(沈苏彬)

guocheng donghua

过程动画(procedural animation)　采用过程来控制动画形体的运动和变化的**计算机动画**技术。在这种动画中,运动由带参数的过程来描述,这些参数可按照一定规律改变,这些规律可以使用解析形式给出或者使用复杂过程(如微分方程组)的解来定义。

最简单的过程动画是按照物理定律(包括运动学、逆运动学、动力学和逆动力学等),用一个数学模型去控制物体的几何形状和运动,如水波随风的运动,物体的变形、弹跳、碰撞运动等。另一类算法动画为粒子和群体的动画。粒子系统是最实用的算法动画技术之一,其基本思想是将许多简单形状的微小粒子作为基本元素聚集起来形成一个不规则的模糊物体,从而构成一个封闭的系统。粒子具有生命周期,它们经历出生、成长、衰老和死亡的过程。粒子系统可成功模拟一些自然景象,如火、烟、喷泉等。

参考文献

1. Parent R. Computer animation: algorithms and techniques. 2nd ed. San Fransisco, CA: Morgan Kaufmann, 2007

2. Kerlow I V. The art of 3D computer animation and effects. 4th ed. Wiley Publishing, 2009

(王裕国　金小刚)

guocheng(hanshu)

过程(函数)(procedure(function)) 一种可以由其他程序调用的、相对独立的代码片段。一个过程(函数)的定义主要包含过程首部和过程体。程序员通常可以在这段代码中说明一些由此过程(函数)专用的变量,称为局部变量。程序的其他部分可以按照其过程首部中指明的信息调用这个过程(函数)。调用时,使用者可以通过参数赋予被调用过程(函数)不同的初始值。每次调用都会从头执行过程体中的代码。一般情况下,函数在执行结束后会回送一个值给调用者,而过程不会回送。

在不同的编程语言中,过程和函数的概念略有不同。一些语言(如 PASCAL)区分过程和函数:函数有回送值,而过程没有回送值。并且调用函数和调用过程的语法是不同的。而另外一些语言(如 C,C ++)把这个机制统称为函数。回送值类型为 void 的函数实际上相当于 PASCAL 中的过程概念。

过程是一种有效的编程机制。一方面,它支持模块化和复用。通过以适当的方式定义和调用过程,人们可以把一个大的程序分解成为多个过程,每个过程可以独立开发。同时,适当定义的过程可以在不同的程序中多次使用,针对一些常用的标准过程,人们还建立了很多标准过程(函数)库,以提高软件开发的效率。另一方面,过程又是一种抽象机制。人们不需要知道具体的实现细节就可以调用它来完成特定的功能。正因为如此,几乎所有的现代程序设计语言都包含了过程机制。

每次调用一个过程时,调用者可以把一些数据传送给过程,使得这个过程完成特定的工作。这种数据传递通常是通过参数传递完成的。在过程的定义/声明中通过形式参数规定了调用者需要传递哪些参数给该过程;调用者必须使用相应类型的实在参数来调用这个过程。不同的语言使用不同的方式来传递参数:

(1) 值传递(ALGOL 60,SIMULA,PASCAL,Ada,C/C ++)。调用者计算好实在参数的值,并把这个值拷贝给形式参数。在过程中对形式参数复制不会影响调用者。

(2) 结果(Ada)。在 Ada 语言中被指明为 out 的参数等同于过程的局部变量。这些参数的实在参数必须是(调用者的)变量。过程中对这些形式参数的赋值操作将会把值赋予相应的实在参数。

(3) 值-结果(FORTRAN,Ada)。形式参数等同于过程的局部变量。其初值为实在参数的值。如果实在参数是变量,那么过程中对形式参数的赋值操作将会把值赋给实在参数。

(4) 引用(FORTRAN,PASCAL,Ada)。实际传递的是实在参数的地址。过程内对形式参数的操作都是通过实在参数的地址完成的。

(5) 名(ALGOL 60)。计算引用实参的子过程 P 被传递到被调用的过程。每次引用形式参数时,过程首先调用 P 得到实在参数的引用,然后再进行操作。

过程也可以把一个值回送给调用者。此时,它们一般被称为函数(在 C/C ++ 中都称为函数)。

参数和回送值是过程(函数)和调用者交换数据的主要途径。过程也可以通过所谓的函数副作用来影响调用者。过程(函数)的执行如果改变了调用者的程序状态,比如某些全局变量的值,就说这个过程(函数)具有副作用。因为很多语言允许在表达式中直接使用函数,而且表达式的各运算分量的计算顺序不确定,函数的副作用可能造成不确定的后果。

如果一个过程的代码(直接或间接地)调用了自身,这个过程就称为递归过程(函数)。递归过程(函数)的好处是表达能力强,且比较简洁,很多时候可以由问题的定义直接得到递归过程(函数)的定义。一些特殊类型的递归过程可以方便地转换成为迭代实现。

在编程中使用过程(函数)的好处是能够有效分解工作量,且有效实现抽象和复用,避免程序中出现过多的重复代码。但是调用过程时,计算机必须执行一些额外的代码来完成参数传递、结果回送、局部变量分配等工作,会造成一定程度的效率下降。为了解决这个问题,有些程序设计语言允许定义内嵌的函数。编译程序会把这些函数的函数体之间嵌入到调用代码中,而不是通过跳转来执行过程。

参考文献

Knuth D E. The Art of Computer Programming, Volume I: Fundamental Algorithms. 3rd ed. Addison-Wesley, 1997

(赵建华)

guocheng kongzhi jisuanji

过程控制计算机(process-control computer) 接受来自受控过程的信息,按一定的控制算法进行处理,及时地作用于过程,使过程得以正常进行的计算机。这里,受控过程主要指的是诸如炼油、化工、造纸、水泥、电力、冶金、纺织等连续性生产过程

以及船舶、飞机的航行过程等。

对于大型工业生产，过程控制计算机是实现安全、高效、优质、低耗的大规模生产的重要保证。过程控制计算机的任务已从测量和控制工艺流程的参数发展到产品质量在线监测与控制，设备运行状态的监测、诊断和事故处理，设备之间的协调控制和连锁保护，生产状态的监控，厂级生产管理决策等。

发展历程

20 世纪 60 年代以后，开始将计算机用于过程控制。大体上经历了 3 个阶段：①20 世纪 60 年代中期以前是试验阶段。50 年代初，在化工生产中首先实现了数据采集和数据处理。由于当时的计算机速度较慢、可靠性差、体积大，没有直接参与过程控制，仅起了对整个生产过程的集中监视，指导生产过程控制及确保生产过程安全的作用。随着计算机运算速度加快和可靠性提高，60 年代初出现了直接数字控制 DDC 系统。在闭环控制系统中，计算机对过程参数进行巡回检测，并依据预先设定的控制参数进行处理，然后发出控制命令直接控制执行机构。②从 60 年代中期到 70 年代初期，随着小型计算机性能价格比的不断提高，在过程控制中的应用得到了很大的发展。但这个阶段仍以计算机集中控制系统为主。在高度集中控制时，为提高控制的可靠性，虽可采用双机工作方式，但投资相应增加，故难以得到推广应用。③随着 70 年代开始的微型计算机的发展，计算机控制也突破了集中控制格局而发展为分级控制。所谓分级控制，是指在受控过程的各个环节直到各项具体装置或设备，都分别用微型计算机实现小范围的局部直接控制，包括一定条件下的局部最优控制。同时采用较为高档的微型计算机系统，通过计算机网络，对所有这些分散的局部控制计算机进行监督控制。如此构成一个二级控制系统。其中起监控作用的计算机称为监控计算机，俗称“上位机”。它依据过程信息和其他参数对过程进行分析和计算，自动地改变给定值，使过程满足某一性能指标（如能耗低、效率高、时间短等）。当局部直接控制计算机出现故障时，还可由上位机完成相应的控制功能，这样可以提高系统整体的可靠性。从 70 年代中期到 80 年代，进一步发展为全分布式控制系统或分布式微处理机控制系统。它综合了计算机、控制、通信和荧光屏显示技术。1987 年提出了体现开放概念的智能自动化系列 I/A Series 和 MOD 300 多回路结构系统。这些系统参考了国际标准化组织推荐的开放系统互连模型，符合制造自动化协议（MAP），方便用户使用和软件移植，为计算机过程控制系统的规范化奠定了基础。

技术特点

过程控制计算机本身的工作原理和常规的计算机没有根本的区别。但针对过程控制的特殊要求，过程控制计算机也有其区别于常规通用计算机的特点，主要有以下几方面：

（1）总线结构　过程控制计算机面向不同行业的受控过程，为满足多方面用户的控制要求，所用的计算机模块和设备应具有良好的兼容性，经不同的组合可配置成各种用途的计算机系统。为此，需要采用标准总线。计算机系统内部总线有 STD 总线、PC 总线等。计算机系统外部总线有 RS - 232C，RS - 422 - A，RS - 485，IEEE - 488 总线等。STD 总线是一种工业标准总线，全部 56 根引线都有确切的定义。STD 总线的特点是模板小型化（模板尺寸为 165 mm × 114 mm）、系统开放式、总线兼容式，可支持 8 位、16 位微处理器。RS - 232 - C 总线定义了机械特性标准和电气特性标准。为提高抗共模干扰能力，RS - 422 - A 总线采用光电隔离 20 mA 电流环接口电路，RS - 232 - C 接口也采用了光电隔离。计算机各部件采用模板化结构，通过总线把各模板连接起来。常用的模板有中央处理器卡、多功能模拟量 I/O 卡、电机控制卡、计数卡、多功能数字量 I/O 卡、光电输入继电器输出卡、信号放大及多路转换卡、接线端子板及安装架、ADAM - 3000/4000/5000 和工业用通信卡 RS - 232，RS - 422，RS - 485 和 IEEE - 488 等。

（2）过程输入输出通道　为了实现对过程的控制，按要求的方式，将过程的各种物理参数送入计算机，计算机经过处理后，将控制命令作用于过程。所以，在计算机和受控对象之间必须设置过程输入输出通道，如模拟量输入输出通道、开关量输入输出通道等。对于实时采集、实时处理和实时控制的系统，还要求在保证精度的条件下，具有有效的实时多路处理功能。

（3）适合于工业生产过程环境的通信网络技术标准　随着现代化工业生产规模不断扩大和控制管理的要求日益提高，在工业生产环境中使用了计算机通信网络。通过它把不同地理位置、不同功能的多台计算机或设备连接起来，达到高可靠性和快速实时响应的目的。为适应工业生产环境，制定了生产过程控制计算机局域网协议，主要有制造自动化协议 MAP 和过程数据高速公路 PROWAY 两种。国

际电工委员会(IEC)于1996年提出了集散控制系统网络的标准体系结构。该体系结构分为3级:第一级为现场总线,第二级为PROWAY,第三级为MAP局域网。它们的功能分别是多点连接众多的可编址I/O装置,连接过程控制器和局部操作站,连接中央操作站、控制管理计算机和生产管理计算机,构成一个计算机集散过程控制系统。

(4)人机接口技术 操作人员和计算机是通过人机接口互通信息的,尽管是在计算机控制下运行,操作人员仍需根据受控过程的状况及时地向计算机发出各种指令,而计算机也应当能实时地显示受控过程状况和操作结果等信息。为此,需要有各种特定用途的外围设备,例如回路操作和显示、键盘和可编程键盘、发光二极管显示器、指示报警装置和荧光屏显示器等。

(5)可靠性技术 在过程控制计算机工作的现场,通常总是存在多种形式的干扰源。因此,过程控制计算机必须强化抗干扰措施,以保证在现场环境中能正常运行。此外,为提高系统的抗干扰能力,还应重视供电技术和接地技术。因为受控过程现场环境往往十分恶劣,例如高温、潮湿、粉尘、振动、化学腐蚀等,所以过程控制计算机必须加强抗震措施、采用抗恶劣环境的机箱等。为确保受控过程的安全和可靠运行,故障诊断和容错技术在控制中的应用越来越广泛。信号报警系统是故障诊断的初级形式,更高的要求是诊断故障的性质、内容和地理位置。容错技术是当系统中某一环节出现故障时,系统仍能维持一定的功能,使系统性能指标保持在一定的范围内。

(6)控制算法及其参数选择和工程实现 按偏差的比例(P)、积分(I)和微分(D)进行控制的PID控制是应用最广泛的常规控制算法。用计算机实现PID控制算法,不仅仅是把PID计算数字化,而且与计算机的逻辑判断功能结合起来,可对标准PID算法进行修正以适应实际的需要。在计算机中,数字式PID控制算法往往编制成公用子程序形式。为共享该子程序,需要给每个PID控制回路配一内存数据区以便存放各个控制回路和输入输出通道的数据。数字PID控制参数的选择,除选择比例增益K_p、积分时间T_i、微分时间T_d和微分增益K_d外,还要选择采样周期T。除PID算法外,尚有参数最优化的低阶控制算法、惯性因子法、达林控制算法和预测控制算法等。现代控制理论的某些高级控制算法在上位机协调控制中已逐步获得应用。同时,智能控制技术,如自适应、自学习、模糊控制、神经网络控制、实时专家控制等,也已获得日益广泛的应用。

(7)系统软件技术 除了程序开发、运行、维护所必需的常规软件环境、工具和相应的支撑软件外,由于在线控制计算机是伴随着受控过程不间断地进行检测、监控,所以必须配备实时操作系统,还应有各种针对性的中断处理程序。此外,针对过程控制的特殊性,除了常规的汇编语言、高级语言外,有时需要采用特定的语言来编制程序。因此,需要有相应的编译程序或解释程序等。

过程控制计算机将沿着上述的微型化、模块化和组合化、开放化、网络化、智能化等方向继续发展。由于过程控制计算机涉及多门学科、综合了多种技术,因此也必将随着相关学科技术的发展而发展。

参考文献

1. 王平,谢昊飞,蒋建春,等. 计算机控制技术及应用. 北京:机械工业出版社,2010

2. 黄德先,王京春,金以慧. 过程控制系统. 北京:清华大学出版社,2011

(李培德)

guochengshi chengxu sheji

过程式程序设计(procedural programming)

使用过程语言设计、编写和测试问题求解程序的活动。过程式程序设计中主要涉及数据结构的确定、求解算法的设计、代码文档的组织和测试等活动。

程序的处理对象是数据,这些数据或者是简单数据元素,或者是由一些数据元素构成的复杂数据,这些数据的逻辑结构统称为数据结构。因此,确定数据结构是过程式程序设计的首要任务。简单数据元素通常有整数、实数、布尔值、字符、枚举值和指针等,复杂数据有字符串、数组、记录、集合、链表、树和图等。为描述数据结构,程序中需要定义数据类型和变量。数据类型可分为简单数据类型和结构数据类型,分别对应于简单数据元素和复杂数据。根据变量在程序中的作用域的大小,变量可分为局部变量和全局变量。局部变量的有效范围只在程序的一小部分,全局变量则在程序的任何地方都可以进行存取。为提高程序的可读性和可修改性,程序中应该尽可能地减小变量的作用域。

求解算法的设计是指利用程序语言提供的控制结构描述求解步骤的活动。过程语言均提供顺序、选择和循环等三种控制结构。顺序结构是指按先后顺序从前到后执行的语句序列。这些语句逻辑上可能有明确的顺序关系,即后一个程序语句依赖于前

一个语句,也可能它们之间没有明确的顺序关系,即某些语句的先后顺序并不重要,一个语句逻辑上并不从属于另一些语句。编写顺序性控制结构时应该把有明确的顺序关系的语句组织在一起,对于没有明确的顺序关系的语句,根据它们对数据的依赖关系,把它们组织在一起,使代码能由上读到下,以提高程序的可读性。**选择结构**是指根据判定条件控制一些语句是否执行的语句。选择结构可用 if-then、if-then-else 或 case(或 switch)等语句进行描述。当判定条件成立时需要执行一组语句,且不成立时不需要执行这些语句,那么应当采用 if-then 语句。当判定条件成立时需要执行一组语句,否则需要执行另一组语句时,可采用 if-then-else 语句。当根据表达式的取值情况在多个动作中选取其一执行时,可采用 case 语句。**循环结构**是指可重复执行一组语句(称为循环体)的程序语句。根据重复方式的不同,循环结构可分为 while 型循环、until 型循环和 for 型循环。while 型循环是在指定的条件(称为循环条件)成立时,重复执行循环体,其特点是执行循环体前先判定循环条件,因此可能一次也不执行循环体。until 型循环将重复执行循环体,直到循环条件成立才结束该重复,其特点是每执行一次循环体后判定循环条件,因此至少执行一次循环体。for 型循环将循环体重复执行给定次数,其特点是循环开始前可确定循环次数。

人们曾提出了多种程序设计方法。自顶向下、逐步求精的程序设计方法是利用抽象机制控制程序设计复杂性的一种典型的程序设计方法。其基本思想是从最能直接反映问题体系结构的概念出发,逐步精细化、具体化、逐步补充细节,直到可用程序设计语言直接描述为止。M. A. Jackson 曾提出了基于数据结构的程序设计方法,他认为程序的控制结构依赖于程序所处理的数据的结构,这一方法给出了根据程序的输入和输出数据的结构来设计控制结构的一套方法。J. D. Warnier 的 LCP 方法是另一种基于数据结构的程序设计方法,这一方法的原理与 Jackson 方法十分相似,但 LCP 方法用更严格的逻辑方法来设计程序。面向目标的程序推导方法是一种比较形式化的程序设计方法,其基本思想是通过从程序的终结断言出发逐步推导最弱前置条件,来推导出相应的程序。

代码文档是程序的最终表现形式。代码文档的表现形式直接影响程序的可读性、可理解性。为此,应该采用好的程序设计风格和适当的注释。程序风格包括好的数据结构、好的命名方法、最小的控制流以及程序文本的清晰布局等。注释是对程序的进一步解释,注释应该描述“为什么”而不是“是什么”。

测试是为发现程序中的错误而执行程序的过程。这里程序的执行具有人工执行和计算机执行双重含义。人工执行是一人或多人为发现程序中的错误,而阅读程序代码的一系列步骤和查找错误的方法,程序审查会、人工运行和复查等是常见的人工执行。计算机执行是在计算机上运行被测程序,并输入一些测试用数据,根据执行结果发现程序中错误的过程。测试方法分为黑盒法和白盒法。黑盒法着眼于程序的外部特征,而不考虑程序的内部逻辑结构。白盒法将根据程序的内部结构,对程序的逻辑路径进行测试。需要指出的是,不管是黑盒测试,还是白盒测试,在实际运用中不可能做到完全的测试,测试只能用于发现程序中的错误,而不能用于证明程序中没有错误。为完成测试任务,需要设计测试用例,其目的在于确定一组有可能发现单个错误或一类错误的测试数据。常用的测试用例设计方法有逻辑覆盖、等价类划分、边界值分析和因果图方法等。

(金淳兆)

guochengshi yuyan

过程式语言(procedure language) 参见**命令式语言**。

guochengxing biaoshi

过程性表示(procedure representation) 一种与推理相结合的**知识表示**方法,知识表示就是求解程序。过程表示是与**陈述性表示**相对应的另一类知识表示方法。

知识的过程表示有两种含义。第一种含义是,把解决一个问题的过程描述出来,称为解题知识的过程性表示;第二种含义是,把客观事物的发展过程用某种方式表示出来,通常应用于理解以自然语言书写的故事,因此称为故事知识的过程性表示。在某些情况下这两种含义很难截然分开。例如,在解决计划制定问题上,解题系统根据用户提出的问题,把求解此问题的过程(解题计划)提供给用户,在此意义上可看作前一种含义;但因为它描述的是一系列事件,所以也可从后一种含义上去理解。

过程性表示方法所表达的知识可以分为两大类:一类是直接与解题有关的知识,它们往往表示

为一套解决某些标准子问题的过程。普通程序设计语言中的标准函数和标准过程都是这种知识;另一类是运用上述知识去解题的知识,即控制知识。表达控制知识的过程,按其表达形式的级别高低,可以分成策略级控制、语句级控制和实现级控制。

采用过程性知识表示的系统,解题的具体过程和算法是直接面向用户的,用户可选用、修改,甚至可以自己设计这些过程和算法。

下面给出一个用过程性表示方法表达知识的例子。假定**知识库**中有如下知识:①人是一定会死的;②狗也是一定会死的;③小张是人;④小李是人。

采用过程表示的知识库由如下过程所组成:

```
Procedure person (x)
  if (x = 小张) or (x = 小李) then return true
  else return false
Procedure mortal (x)
  if person (x) then return true
  else if dog (x) then return true
  else return false
```

在这个知识库中,如果想要询问小李是否一定会死,则只需执行过程 mortal(小李)就可得出结论。

通过例子可看出,采用过程性表示的知识库具有如下特征:它是一组过程集合,对它的修改是通过增加、删除或修改过程来完成的。

过程性表示的优点:①有利于表示启发式知识和默认推理知识;②能实现扩充逻辑推理(如非单调推理、不精确推理等);③不需要**推理机**,执行效率高。

过程性表示的缺点:①由于知识隐含在过程中,因此难以修改和证明,表达不够严格;②固定的控制信息限定了其他可能的方法。在一个过程中,由于问题求解活动隐含的方向性,使得一些本没有次序的解题方法添加了不必要的隐含次序控制信息;③由于过程性表示依赖于任务,所以通用性较差;④采用过程性表示的系统使用了启发式推理知识,这些知识本身是不完备的,有时在条件充分的情况下也未必能得出最优答案,有时还不能保证推理过程是正确的。

参考文献

1. 陆汝钤. 人工智能(上). 北京:科学出版社,1989

2. 管纪文,刘大有,等. 知识工程原理. 长春:吉林大学出版社,1988

(刘大有　唐海鹰)

guocheng yuyan

过程语言(process language)　一类描述动作过程的语言。“过程语言”作为独立术语的用法并不多见,出现更多的如“过程建模语言”(process modeling language),“过程规约语言”(process specification language)等,特别是目前影响较大的 WS-BPEL(Web service business process execution language,Web 服务业务过程执行语言),都可以认为是过程语言。过程语言研究的一个起因是工作流(workflow)领域研究和应用的发展,因此可以看作是工作流应用需求的总结和抽象。另一方面,近年 Web 应用和 Web 计算的兴盛也极大地推动了过程语言的发展,因为基于 Web 服务的系统就是利用一些已有服务(作为基本动作)建立相应的执行过程,并需要描述过程执行中的相互联系(包括数据传递等)。过程语言的设计也受到进程代数(process algebra)领域理论研究的深刻影响。过程语言的设计,从根本上看,就是为了灵活方便地描述一些基本动作形成的工作流程。这里通常并不关心基本动作具体做什么,而且有关工作流程可能很复杂,存在实际应用需求提出的许多特殊约束和控制需求,其中许多需求在常规的编程语言中不能得到很好的支持。有些控制结构与常规的顺序或并发编程语言中的控制结构类似,如业务过程中的动作可能有特定的顺序约束,或者需要同时执行的动作和适当同步等。也有很多告知机制在常规编程语言中没有直接的对应,如可能要求前后或同时执行的动作之间具有复杂的选择性约束关系,在某些情况下要求撤销已经(可能是部分)完成的动作(回滚),定时进行的动作和超时后的动作撤销和替换,事件驱动的动作和过程性动作的融合等。这种基于基本动作描述流程从而建立所需应用系统的过程也被称为“编制”(orchestration)。

参考文献

1. Business Process Model and Notation, http://www.bpmn.org/Documents/BPMN V1-0 May 3 2004.pdf

2. Web Services Business Process Execution Language, version 2.0, 11 April 2007, http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html

(裘宗燕)

H

hanshushi chengxu sheji

函数式程序设计(functional programming)　编写函数式程序的方法与过程。其主要任务在于定义(或构造)函数以求解所提出的问题。定义的函数(可看作主程序)可能包括一些辅助函数(可看作子程序)。计算机按照所定义函数的相应表达式,根据计算规则逐步计算,最后得到所需的结果。表达式中可能包含函数名,计算时可将其相应的定义作为归约规则。

函数式程序设计的一个显著特点是:若一个表达式有定义,则该表达式的最后结果与其计算次序无关。函数式语言可分为纯函数式语言与非纯函数式语言(参见**函数式程序设计语言**)。纯函数式语言中的变量与数学中的变量一样,它们只代表函数的参数值,在整个表达式的计算过程中其值始终不变,称为**引用透明**。基此,表达式可按照某些代数定律进行等式变换和推理。

函数式语言在表达能力方面还具有两个特点:一为构造数据的能力,即把整个数据结构看作简单值,它们可作为参数值被传送,也可作为计算结果被回送。一旦被建立后就不能改变。它们可纳入到其他结构中去,但只要需要它,其值始终有效。这使得函数式语言便于使用,但在实现时为了提高效率(特别是空间的使用效率),提出了许多研究课题。二为建立高阶函数的能力,其参数可为函数,或其结果可为函数的函数称为高阶函数。使用高阶函数可使程序简洁、清晰,但其正确实现是实现技术中的研究课题。上述两个特点构成函数式程序设计的概念架构。它既简洁、精巧,又富有灵活性和表达能力。

与使用一般程序语言相比,函数式程序设计存在如下问题:①由于函数式语言中不允许赋值语句,故无副作用,因此在表达诸如输入/输出,进程间的相互作用等方面就比较困难。虽然已有一些进展,例如使用 Monad,但离解决问题还远;②运行效率还亟待提高。究其原因,主要在于函数式程序设计语言的计算模型为λ演算,而目前的冯·诺依曼计算机是以**图灵机**为计算模型的。根本的解决途径是寻求一种适合于函数式程序设计语言的全新的计算机体系结构。

参考文献

1. Bird R, Wadler P. Introduction to functional programming. Prentice Hall, 1988

2. Jones S P. The implementation of functional programming languages. Prentice Hall,1987

(孙永强　黄林鹏)

hanshushi chengxu sheji yuyan

函数式程序设计语言(functional programming language)　用于**函数式程序设计**的语言。其中函数是构造程序的基本成分,并提供一些设施用于构造更为复杂的函数。程序人员根据提出的问题去定义求解函数(即为主程序),其中可能包含一些辅助函数(即为子程序)。最早的函数式程序设计语言是 J. McCarthy 领导的小组所建立的 LISP-1 语言。1977 年,J. Backus 在他的 ACM Turing 演讲中,详细、深刻地分析了冯·诺依曼风格的语言——FP 函数式语言。他的演讲引起了世人的广泛注意,被称为函数式语言发展史上的里程碑,其后陆续出现了一些新的函数式语言如 SASL, KRL, ML, Miranda, Hope, Orwell, Haskell 等。

函数式语言可分为纯函数式语言与非纯函数式语言。纯函数式语言中禁止使用赋值语句,从而不会产生副作用,其优点是享有引用透明性,有助于程序的等式变换和推理。非纯函数式语言中,允许赋值语句在一定范围内使用,在沿袭函数式语言所享有的优点的同时,提高表达能力和实现效率。按**语义**区分,又可分为严格的和非严格两类。一个函数 f 称为严格的,当且仅当 $f(\perp)=\perp$。在严格函数式语言中,参数是按值调用,这种计算方法称为急性计值,反之,非严格函数式语言中,参数的计值是惰性的,参数只在需用时才计算,目前有许多非严格的函数式语言,P. Hudak 等人组织了一个小组,定义了 Haskell 语言,企图作为这类语言的标准文本。

参考文献

1. Bird R, Walder P. Introduction to functional programming. Prentice Hall,1988

2. Jones S P. The implementation of functional programming languages. Prentice Hall,1987

（孙永强　黄林鹏）

hanyu xinxi chuli

汉语信息处理（Chinese language information processing）　利用计算机对汉语所承载的内容信息进行加工、操作的理论、方法和技术。自然语言通常有书面语言和口语之分，这里只阐述用**汉字**记录的书面汉语的处理（口语信息处理请参见**汉语言语识别**和**汉语言语合成**）。在**汉字信息处理**阶段，关注的是汉字的字符（一种图形）信息，而在汉语信息处理阶段则关心汉字所携带的内容信息，要把处理对象——汉字串区分为不同的语言单位：词、短语、句子、篇章或话语乃至文献集，要在词法、句法、语义和语用等不同的层面上进行研究。汉语信息处理既包括分析汉语（参见**自然语言分析**），也包括生成汉语（参见**自然语言生成**）。汉语分析的目的是让机器能“读”或“理解”汉语，汉语生成则是让机器能“写”或“运用”汉语。汉语信息处理的重要应用领域，如**机器翻译**和跨语言信息检索。它们是涉及多种语言的，因此，汉语信息处理实际上就是以汉语为核心的多语言信息处理。

实际上，**数字计算机**在非数值计算领域中的应用最早是在自然语言处理领域内开始尝试的。计算机问世不久，就开始了机器翻译试验。但无论与计算机技术本身还是与其应用技术的飞速发展相比较，自然语言处理的进展都是相当缓慢的。自然语言处理面临重重困难，首当其冲者是自然语言的歧义现象。人与人用自然语言进行交流并没有十分明显地遭遇歧义的困扰，这是因为交流总是在一定的环境中进行的，交流双方的知识背景有共同部分，而且交流的目的大体上也有了预设，因此自然地化解了自然语言的潜在歧义。当计算机程序分析自然语言文本时，自然语言的歧义便凸现出来。

汉语分析可分解为词法分析、句法分析、语义分析、语境分析等步骤。在不同的步骤中，需要消解不同的歧义问题。汉语词法分析的重要任务是词语切分或辨识。由于在中文语句中汉字是连写的，呈现的是汉字串，词与词之间没有空格，词语切分成了汉语自动分析的首要步骤。这一步主要解决的是切分歧义。先看以下例句。

　　她将来会成为一位科学家

　　白天鹅飞过来了

人不难把它们正确地切分为：

　　她　将来　会　成为　一　位　科学家

　　白　天鹅　飞　过来　了

计算机采用基于词典（假设该词典收了“她”、“将”、“来”、“将来”、“会”、“成为”、“一”、“位”、“科学家”、“白”、“天”、“白天”、“天鹅”、“鹅”、“飞”、“过来”、“了”这些词，但没收“白天鹅”）的从右到左最长匹配算法也能实现如此切分。但是，当碰到以下句子时，

　　他将来北京进修

　　白天鹅可以看家

得到的结果（假设该词典中还有“北京”、“进修”、“可以”、“看”、“家”这些词）是

　　他　将来　北京　进修

　　白　天鹅　可以　看　家

由此进行后续分析必然是错误的。这里“将来”有两种可能的切分：①作为一个词，②作为两个词；“白天鹅”也有两种可能的切分：①“白 天鹅”，②“白天 鹅”。可以设计一种算法，将这两种可能的结果都发现出来，但如果要求计算机能从中选择出正确的，就要预先教给计算机一些语法和语义知识。如：

（1）作为一个时间词，“将来”是不能直接修饰名词“北京”的。

（2）“天鹅”是会飞的，而家禽“鹅”是不会飞的。

（3）在南方农村，“鹅”可以看家，“天鹅”不会看家。

如何形式化地描述这些知识并将其组成计算机便于利用的知识库，成为自然语言处理系统的基础课题。如果处理句子“白天鹅在湖里游泳”，“白天鹅”的两种切分都是成立的，这时仅利用知识库中的静态知识还不够，还要利用上下文中的动态语境知识，以判断在湖里游泳的是“天鹅”还是“鹅”。由此可见，自动消解歧义得到正确的分析结果并不是一件容易的事。人们为此已经付出了几十年的努力。

长期以来，面向文本的自然语言处理研究的主攻方向是歧义消解。然而消解了歧义还不能说就实现了理解。自然语言运用中经常使用的隐喻、影射、双关、夸张、幽默、拟人等表现手法以及遣词造句的

技巧对自然语言处理提出了新的挑战,识别与理解隐喻就超出了歧义消解的范围。还需要认识到,现在的自然语言处理研究只以文本为对象,局限性是明显的。实际上人类阅读与交际是多通道的,脑、口、眼、耳并用,实现语音、文字、图像多模态信息的融会贯通。目前的自然语言处理研究才刚刚认识到这一点。实现自然语言处理到自然语言理解的突破必须仰仗计算机科学、语言科学、脑科学、认知科学的共同进步,多学科的交叉和融合才有希望。

参考文献

1. 俞士汶. 计算语言学概论. 北京:商务印书馆,2003

2. 宋成庆. 统计自然语言处理. 2 版. 北京:清华大学出版社,2013

(俞士汶)

hanyu yanyu hecheng

汉语言语合成(Chinese speech synthesis)

用机械、电子、计算机等人工方法生成汉语言语的过程与技术。用于生成汉语言语的计算机系统叫做**语音合成器**。语音合成器可以利用软件或硬件实现。文语转换系统(参见**文语转换**)可以将语言的文字转换为语音。合成语音可以通过拼接存储在语音数据库中的录制好的语音合成单位**语音基元**产生。合成系统因存储的语音基元大小不同而有所差异。一个通用语音合成系统可以通过存储音素或者双音素来获得各种文字的合成语音,但合成语音可能不够清晰。而一个面向特定领域的系统可以通过存储词或者语句来获得高质量的合成语音。此外,一个合成器也可以通过与声道模型以及其他人的语音特性相结合来合成出不同的语音。合成语音的质量可以通过与自然语音的相似程度以及可理解程度衡量。言语合成有着广阔的应用前景,它可应用于盲人计算机、电话信息查询、文本校对、专家系统的有声输出、机场航班信息报告等领域,它也是汉语人机语音通信和智能计算机接口必不可少的组成部分。

1939 年,贝尔实验室 Homer Dudley 制作的第一个电子合成器 VODER(voice demonstrator) 在美国纽约世界博览会上展出。这是一个利用共振峰原理制作的语音合成器。VODER 用手指按十个琴键,控制带通滤波器;三个额外的琴键控制爆破音;手腕控制声源开关;脚踏板控制张弛振荡器以改变声调,从而产生各种声音。

1960 年瑞典语言学家和言语工程学家 G. Fant 在论文"Acoustic Theory of Speech Production"中系统地阐述了语音产生的声学理论,推动了语音合成技术的进步。线性预测分析是最有效的语音分析技术之一,利用这个技术可以对语音产生模型的参数进行准确估计。20 世纪 70 年代以后,线性预测技术开始用于语音编码和识别。同时可以根据线性预测参数用多种方法合成语音。

1980 年,美国麻省理工学院 D. Klatt 教授设计了串/并联混合型共振峰合成器 MITalk 系统。它用串联通道产生元音和浊辅音,并联通道产生轻辅音,还可以对声源作各种选择和调整,以模拟不同嗓音。在此基础上开发的 DEC Talk 英语文语转换系统获得了广泛应用。

20 世纪 80 年代末,E. Moulines 和 F. Charpentier 提出了基于时域波形修改的语音合成算法 PSOLA (pitch synchronous overlap add),该方法较好地解决了语音拼接中的问题,从而推动了波形拼接语音合成和文语转换技术的发展和应用。

20 世纪 80 年代和 90 年代主要的合成系统为 MITalk 系统和贝尔实验室的语音合成系统。贝尔实验室的系统是第一个多语言语音合成系统,广泛使用了自然语言处理技术。

20 世纪 90 年代末,日本的 Tokuda 教授提出了基于隐马尔可夫模型(hidden Markov model,HMM)的语音合成方法,该方法也叫做统计参数语音合成。在这种方法中,语音的频谱、基频和时长同时用隐马尔可夫模型建模,利用最大似然准则,通过隐马尔可夫模型产生合成语音。

汉语言语合成的研究始于 20 世纪 50 年代。20 世纪 80 年代,我国学者设计了以汉语普通话声母、韵母为合成单位的共振峰合成器。20 世纪 90 年代初,国内的汉语 TTS 系统逐渐转向波形拼接算法。

言语合成采用的技术(参见**言语合成方法**)可分为发音器官参数合成、声道模型参数合成、波形拼接合成和统计参数合成;合成策略可分为频谱逼近和波形逼近。发音器官参数合成通过直接模拟人的发音过程来合成语音。声道模型参数合成方法通常采用言语产生的源滤波器模型来合成语音,以共振峰合成为代表。它的优点是占用的存储空间小,与言语编码相结合时数码率较低,同时合成言语具有较高可懂度,能够较灵活地控制合成言语的音色,但不够自然。波形拼接合成通过对来自自然言语的语音基元进行拼接产生合成语音,具有较高的自然度,以基元选取语音合成为代表。这种方法需要占用大

量的存储空间,合成言语的音色相对固定。统计参数合成通过对语音参数建立统计模型,进而通过最大似然准则,利用统计模型产生语音。这种方法结合了参数语音合成与基元选取语音合成的优点,能够获得较高自然度的合成语音。

参考文献

1. 蔡莲红,黄德智,蔡锐. 现代语音技术基础与应用. 北京:清华大学出版社,2003

2. Klatt D. Review of text-to-speech conversion for English. Journal of Acoustic Society of America, 1987,(82)3:737-793

(蔡莲红 杨鸿武)

hanyu yanyu lijie

汉语言语理解(Chinese speech understanding) 让计算机理解汉语自然口语的技术。计算机不仅可以识别出连续言语的内容,而且可以像人一样理解人们说话的意思,回答人们提出的问题,或按人们的意图去执行特定的操作。**自然语言理解**可分为口语理解和书面语理解两方面,这里指的是口语理解。口语发音常出现习惯性无意义添加词、词序颠倒、重复、停顿等现象,因此口语理解要有提取关键词的功能,这和规范的文本理解有较大差别。

汉语言语理解和英语、欧语相比,更为困难。首先汉语词间无间隔标记,汉语词汇自动切分本身就是一个较难的课题,由此引起的歧义现象更为严重。汉语还有连动句和兼语句两类特殊句型,再加上汉语量词丰富、成语众多,使汉语理解更难于用机器实现。基于以上原因,汉语言语理解研究目前也仅局限在极小应用范围的表演系统,而且只是发音规范的特定人连续言语识别和理解系统。例如一个词汇表为89个词的火车订票和信息查询连续言语理解系统,其涉及的词及由此组成的合法句子都局限在较小范围内。言语理解是人机口语对话和机器口语翻译的基础。

包括中国在内的20多个国家参与的国际组织“言语翻译先进研究协会”,提出了七国电话言语同声翻译研究计划(C-STAR Ⅲ),是言语理解应用最宏伟的长期研究计划。

参考文献

1. Waible A, Lee K F. Readings in speech recognition. San Mateo: Morgan Kaufmann Publishers Inc., 1990

2. 刘开瑛,郭炳炎. 自然语言处理. 北京:科学出版社,1991

(莫福源)

hanyu yanyu shibie

汉语言语识别(Chinese speech recognition) 由**计算机系统**对汉语言语所携带的信息进行分析与感知的过程和技术。

发展简史

言语识别(也称语音识别)的历史可以追溯到20世纪50年代。1952年K. H. Davis用电阻、电容、电子管等分立元件,实现带通滤波器组进行语音信号频谱分析和匹配,10个阿拉伯数字的识别率达98%。1960年P. Denes等研究成功第一个计算机言语识别系统,开创了计算机言语识别的新阶段。同年,瑞典科学家G. Fant提出了著名的言语产生的声源——滤波器模型,奠定了现代语音信号处理的理论基础,对言语分析、合成和识别工作起了巨大的推动作用。20世纪60年代中期,线性预测技术引入语音信号处理,提供了较为精确的声道响应估算办法,从而引出了LPC、LFCC、LPCC、LSF等一系列语音特征。十年后,人们按照人耳对音高感知的非线性特征,推算出比线性预测特征更好的MFCC特征。70年代初,动态时间规正匹配算法较好地解决了语音特征时域上非线性变化问题,提高了识别率。70年代中期,J. K. Baker等人将隐马尔可夫模型(HMM)应用到言语识别领域。由于HMM既可描述语音的瞬态特性,又可描述语音的动态转移特性,合理地反映了语音的统计特征,在言语识别中获得极大成功,已成为当今言语识别的主流算法。1975年前后,Dragon第一次实现了真正意义上的连接词语音识别系统。该系统可以识别的词汇量大约1000个,且句子中的每个词间都需要有小停顿,识别一句话需要一个小时。

在随后时间内,研究人员对HMM模型作出了许多卓有成效的改进。高斯混合模型(GMM)是当时最成熟的统计概率模型,最早用于构建HMM的每个状态。人们对HMM模型的改进也集中在对GMM的参数估计上。起初,人们将多个人的录音放在一起训练模型,形成与说话人无关的识别器,实现了从说话人有关到说话人无关的突破。随后,最大后验概率(maximum A posteriori, MAP)和最大似然线性变换(maximum like lihood linear transform, MLLT)等自适应技术的提出使得通用模型可以针对每个人的特点进行均值、方差、权重的调整,进一步提高了识别率。此时的HMM模型的训练目标是使得模型在训练集上的似然概率最大化,而识别的目标是使得从给定数据观察到模型序列的错误率最小

化。训练和识别的这种不匹配激发人们进一步探索,提出了 MMI、bMMI、MPE 等一系列区分度训练(DT)算法,从而使识别错误率进一步下降。

在 90 年代初期,人们发现采用**人工神经网络**的输出作为 GMM 模型的输入可以获得性能上的提升,这种模型被称为 TANDEM 模型。除此之外,研究人员采用神经网络代替 GMM 模型,也取得比 GMM 模型更好的效果,这种模型被称为 Hybrid 模型。神经网络性能虽好,但计算复杂度较高,在当时的硬件条件下无法对大规模语料进行训练。直到最近几年,高性能显卡的出现将训练时间减少了两个数量级,Hybrid 模型才真正得以流行起来。2009 年,采用约束玻耳兹曼机及约束玻耳兹曼机(restricted Boltzmann machine,RBM)理论构建而成的深度神经网络(DBN-DNN)在言语识别领域取得重大突破,其构建的单音子模型以错误率下降 30% 的绝对优势打败了经过区分度训练优化后的 GMM-HMM 三音子模型。随后,深度神经网络受到高度重视,在模型快速训练、快速解码、错误率优化三个方面取得若干重大突破。

汉语言语识别研究工作始于 1958 年,中国科学院声学研究所用电子管设备识别 10 个元音。1972 年起该所开始进行计算机言语识别。20 世纪 70 年代末以来,有更多单位,先后开展了汉语言语识别的研究,现在已有近百个单位参与这方面的研究工作。总体来说,汉语言语识别技术与英语言语识别技术没有本质区别。许多研究人员一直处在跟进国外先进技术的状态,并对先进技术作了一些小改进,源头创新依然掌握在国外主要研究机构手中。

汉语言语(语音)的特点

汉语属于汉藏语系中的汉语语族。汉语语族包括七种方言:北方方言、粤方言、闽方言、客家方言、吴方言、赣方言、湘方言。各种方言在语音、词汇、语法方面存在不同程度的差别。为使汉语表达走向规范化、标准化,国务院号召推广使用“普通话”。普通话语音音系比较简单,音节的结构形式也比较少;从听感上会觉得清亮、高扬、舒缓、柔和;在口头运用中,有鲜明的轻重音和儿化韵等变化。

汉语发音通常以一个汉字作为一个音节。语言语音体系的基本元素是元音和辅音。中国传统的音韵学研究把汉语语音分为声母和韵母。把每个音节分成两部分:字音的前一部分称为声,后一部分称作韵。辅音可以作声母,但不是所有的辅音都能做声母。元音可以做韵母,但韵母部分可以包括不止一个元音,也可以包括辅音,如 ng[ŋ]。

汉语普通话包括 64 个音素。音素构成了 21 个声母和 38 个韵母。声母(可以没有)和韵母组合成音节。汉语普通话以北京话为基础,其声母和韵母的组合基本上与北京话相同,共有 400 多个无调音节,1600 多个带调音节。它没有尖团差异,没有颤音拍音,没有清浊对立,没有松紧对立。

汉语是声调语言。汉语同声同韵字较多,赵元任以《石室诗士食狮史》说明了声调对这部分音节的区分具有重要作用。早在南北朝时期,汉语便分“平上去入”四声。唐宋时期又按照声母清浊分为阴阳调,共计八调。元朝时期随着外族入关四声八调格局被打乱,出现“平分阴阳、入派三声”。《中原音韵》将其归类为“阴平、阳平、上声、去声”,被现代汉语普通话所沿用,称为“声调”,且引入一个特殊的轻声调。刘复在《四声实验录》中断定,四声的差别在于乐音的高低差异,并由此描绘记录了 12 种方言单字调的调型曲线。汉语的每个音节都有自己稳定的静态声调,称为单字调。在连续语流中,声调调值发生或多或少的变异,主要分为三种:第一种是受到上下文声调影响产生的自然变异。它通常不被人们所感知,只有通过仪器测量才能发现。第二种是受到语调影响而产生的变异。它有一部分可以被人们明显感知(如无标记疑问语调末音节),另一部分不被人们明显感知(如语调下倾导致的平声不平)。第三种是受到约定俗成的变调法则约束的变异。这部分变调法则比较复杂,各个方言区各不相同。汉语普通话较为简单,只有“上声相连前字变阳平”的变调法则。而闽南方言则较为复杂,每个汉字均有单字调和前变调,通过前变调法则可以严格区分韵律词边界,从而解决汉语普通话所不能区分的“东四/十条”和“东/四十/条”的韵律边界,除此之外每个韵律句字末字还可以应用后变调法则,表示该韵律句的语义还未表达完整,需要与后面的韵律句合并起来才能完整表达意思。各种形式的声调和变调,给汉语言语识别带来了一定困难,把握声调的规律,可以有效地提高言语识别性能。

基本原理

各语种言语识别的基本原理大致相同。框图如图 1 所示。

言语识别系统实现的过程分训练和识别两个阶段。训练阶段是在机器中建立被识语音的样板或模型库,或者对已存在机器中的样板或模型作特定发音人的适应性修正。在识别阶段,将被识语音信号

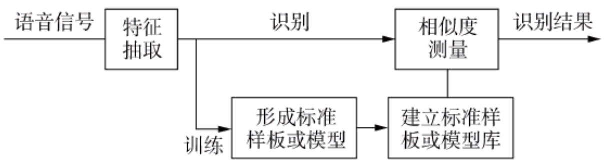


图1　言语识别的基本原理框图

的特征参量提取出来（参见**言语识别的特征提取**）进行模式匹配，相似度最大者（或距离测量最小者）即为识别结果。在大词汇、连续言语识别和口语理解的情况下，使用语言模型对提高识别速度和正确率起到很大作用。汉语言语识别中，考虑汉语的特点会对汉语识别有所帮助。首先汉语是以音节为基础的语言，它们都是元音结尾或元音加鼻韵尾的开音节结构；汉语协同发音和音变不如英语严重；汉语是有调语言，“调”起辨意作用等。这些特点在汉语识别中可资利用。

言语识别分类

1. 按照解码算法划分，言语识别可以分为基于模板的方法和基于模型的方法。基于模板的方法需要保留一些标注好的录音作为模板，通常采用动态时间弯折（DTW）及其变种对待识别录音和标注过的模板音进行匹配，选择最接近的模板的标注作为识别结果。基于模型的方法通常采用HMM模型作为声学特征的概率模型，采用多元文法模型、发音词典、音子上下文模型约束解码空间，采用维特比解码算法作为基本解码策略，并采用集束搜索（Beam Search）辅助快速剪枝。

2. 按识别对象划分，言语识别有以下3类识别方式。

（1）孤立词识别　孤立词识别是指在发待识别的音时，被识单元间有明显的停顿。识别系统不需要特殊处理来分割单元。对汉语而言，以字、词或短语为单元构成词汇表，待识语音则为这些单元中的某一个。

按词汇表大小，可分为小词汇表（100词以下）、中词汇表（100～1000词）和大词汇表（1000词以上）3类。中、小词表识别可用整词作为识别单元，此即孤立词识别。由于大词表的混淆性大大增加，只能用子词单元（如音素、声韵母、音节等），识别难度很大。

（2）连接词识别　连接词识别的识别词汇表也是字、词或短语，但识别时可以是它们中间几个的慢速连读。慢速连读是指既不像孤立词识别时单元间有明显间歇，也不像连续言语识别那样需用复杂的程序来切分单元，而是机器仍很容易分割单元。识别时只会产生替代错误，不会产生插入错误和丢失错误。

（3）连续言语识别　连续言语识别的待识语音信号是完整的句子，以正常说话速度发音，比连接词发音要快得多，甚至还允许有一定的随意性。句中每个字或词与它们单独发音比，除了有字调和词调的变化外，由于它们在句中所处位置不同，受整句语调的影响。识别单元可以字或词为单元，也可用声母、韵母等音素为单元。由于上述原因，连续言语识别要对识别单元正确切分和识别，是言语识别主要难题之一。

3. 按使用者适应情况划分，言语识别可分为以下两类。

（1）特定人言语识别　特定人言语识别又称为认人言语识别，识别系统只适应某一个或某一些特定人。系统训练时只用使用人的语音来生成识别单元的标准样板或模型（参见汉语言语识别）。因而仅适合其本人或本组人使用。其他人使用时要重新训练以产生适合自己的标准样板或模型，否则识别率将大为下降。

（2）非特定人言语识别　非特定人言语识别又称为不认人言语识别，识别系统在不加特殊训练的情况下，可适应相当范畴的说话人（例如说普通话的人），而不是仅适应某个或某些特定人。这种系统显然需要由一些属于这一范畴的发音人来训练标准样板或模型。该系统可供参加训练者（圈内人）使用，也可供未参加训练者（圈外人）使用。由于不同讲话者之间不仅发声器官（声道长度、鼻腔大小等）有差异，而且发音习惯如口音、速度、响度等也有所不同，因而严重影响识别效果，使不认人识别也成为言语识别难题之一。

应用领域

言语识别技术在许多领域都有广泛的应用，在国防安全、家居生活、信息检索等领域均有许多用途。采用言语识别技术可以对监控录音进行关键字搜索，防止出现有害国家安全的字眼；在汽车运行期间，可以采用语音控制汽车的音响门窗；在日常生活中，可以采用语音操作电视，控制窗帘等设备；在信息检索时，可以检索视频和音频中说过的话语内容。言语识别技术的应用场景十分广阔，还有许多领域等待我们进一步开发。

参考文献

1. 陈永彬,王仁华. 语音信号处理. 合肥：中国科学技术大学出版社,1990

2. Waibel A, Lee K F. Readings in Speech Recognition. San Mateo: Morgan Kaufmann Publishers Inc., 1990

3. 方棣棠. 汉语语音识别的当前任务与研究方向. 见：第三届全国人机语音通讯学术会议论文集. 成都：四川科学技术出版社,1994

4. Li Jing, Zheng Fang, Zhang Jiyong, et al. The definition and extension of the question set for decision tree based state tying in Chinese speech recognition. International Conference on Chinese Computing. Singapore, 2001, pp. 106-110

（徐波　柯登峰　莫福源　方棣棠）

hanzi

汉字（Hanzi, Chinese character, Han character, Chinese Hanzi）　主要用于记录汉语（我国台湾地区习称国语,新加坡、马来西亚等地亦称华语）的文字。

汉字属于表意文字。相对于表音文字（或拼音文字）而言,表意文字是用符号直接表达词或词素。在信息技术的国际标准中,往往用“表意文字（ideographic character）”泛指汉字,而用 Chinese Hanzi/Japanese Kanji/Korean Hanja 特指在中国、日本、韩国所使用的汉字。

我国汉字自古至今累计总数已超过 70 000 字,但现代汉语常用字的数量远少于此。国家语言文字工作委员会 1988 年颁布的《现代汉语通用字表》包含 7000 字,国家标准局 1981 年发布的 GB 2312—1980（《信息交换用汉字编码字符集 · 基本集》）收录汉字 6763 个,2005 年发布的国家标准 GB 18030—2005（《信息技术 中文编码字符集》）共收录汉字70 244 个。

在信息处理中,汉字的字形、字音、字义及由此派生的一些属性起着十分重要的作用。

汉字的字形　汉字字形是汉字形体结构的图像,是汉字的表现形式。除了极个别的特例,汉字都是方块型结构。汉字字形属性包括以下内容：

（1）部首　汉字形体偏旁所分的门类。我国以 201 部首为现代汉字部首的推荐规范。国际上习用《康熙字典》的 214 部首。在对汉字排序时,如果部首是排序的因素,则每一个汉字的部首必须唯一。

（2）部首外笔画数　一般与部首联合使用,用于排序和（或）检索。

（3）总笔画数。

（4）结构特征　指一个汉字形体的左右结构、上下结构、包围结构和镶嵌结构等。

（5）汉字笔画序列　按照汉字的规范书写顺序,将汉字拆分为横竖点撇折（— 丨 丶 丿 乙）等笔画的序列。

汉字的字音　汉字字音源于汉语读音。除了极个别的特例,一个汉字只有一个音节。汉字读音由 400 多个无调音节构成。加上四声变化,共有 1290 多个带调音节。汉语拼音采用拉丁字母和一些附加符号表示汉语普通话的发音,一般采用小写拉丁字母,并通过使用声调附标“ˉ”、“ˊ”、“ˇ”和“ˋ”分别表示四个不同的声调（即第一声阴平、第二声阳平、第三声上声、第四声去声）。

汉字具有较强的多字同音和一字多音性质,字音与其字义密切相连（与词语背景相关）,在**汉语言语识别**、**汉语言语合成**等处理过程中起着重要的作用。

汉字的字义　每个汉字的含义一般常有 2—5 个,有的多达 6—9 个,少数字甚至有十几种不同的意义。汉字的字义需根据它们的上下文进行分析才能确定。汉字又具有很强的构词能力,复合词（由 2 个或多个汉字所构成）数以万计,这就给汉字文档的自动分类、自动摘要、关键字提取、文档检索、自动翻译等计算机处理带来了许多困难。

汉字的排序　汉字与拼音文字不同,拼音文字由字母组成,而字母有固定的排序,因此拼音文字的排序和检索相当方便。但汉字是表意文字,数量庞大,没有统一、固定的排序方法。目前常用的汉字排序方法有 3 种：

（1）按部首排序　部首是将汉字里常见的共同偏旁,如金、木、水、火、土、人、鬼等,拿来作为分类汉字的基准,所有汉字都被归类在某个部首中。部首是现代汉语词典最主要的检索方式之一。由于部首只能作大略分类,因此还需搭配笔画检索。通常的排序方式是先依部首的笔画数排序,部首相同的字再依去掉部首后的剩余笔画数排序。同笔画数的部首或同剩余笔画数的字,并无统一的排序标准。大多数字典会沿用《康熙字典》的排序,但也有些字典依照自己的理念加以调整。部分汉字归属哪个部首也有类似情况。

（2）按笔画数目排序　根据汉字的总笔画数排

序。同笔画的汉字没有统一的排序标准,通常会再按部首、读音、字形、字义等方式做次级排序。由于笔画相同的汉字数量很多,因而检索效率不高,目前多用于书籍后的术语索引。字典由于数据量太大,读者较少使用笔画检索,但字典多会附上笔画检索表,作为替代的检索方式。

(3)按读音排序　按照汉字拼音的字母顺序进行排序,拼音字母相同时按四声排序,同音同声的汉字按笔画数排序,笔画相同时再按起笔笔形(横、竖、撇、点、折)的顺序排列。按读音排序是目前采用较多的一种方法。

汉字的字体(typeface)　在书法和印刷领域中字体是指文字的风格式样。常用的汉字字体有几十种,如宋体、仿宋、黑体、楷体、圆体、魏碑、隶书、行书等。

目前计算机中的汉字大多采用一组直线段或曲线段描述其笔画的内外轮廓线,这种描述方法能很好表示各种字体的汉字,其字形质量高,描述数据量小,便于无级变倍和字形变化。一些便携式数字设备考虑到成本和用途等因素,采用 $m \times n$ 的点阵(如 16×16,24×24,32×32 等)表示汉字,字体风格难以表现,字号大小变换时字形质量无法保证。

计算机中用同一方法制作的某种字体所有字符(汉字)的全部字形描述数据称为 font,电子出版领域称为字模,非专业用户也称为字型或字库。

参考文献

中国社会科学院语言研究所词典编辑室. 现代汉语词典. 5 版. 北京:商务印书馆,2008

(张福炎　张轴材)

hanzi bianma zifuji

汉字编码字符集(Coded Chinese Character Set(狭义),Coded Ideographic Character Set(广义))　为处理汉字信息而按照一组无歧义的规则所定义的**汉字**和其他字符的集合,其中每个汉字和符号都有其唯一的代码表示。汉字编码字符集用于汉字信息的表示、处理、交换、传输与存储。

汉字是大字符集,每个汉字至少需要用两个字节来表示。目前在计算机信息系统中使用的汉字编码字符集有多种,大体可以分成两类:一类是以汉字作为主体的汉字编码字符集;另一类是包含世界各国和地区使用的所有文字符号的多文种编码字符集。

1. 以汉字为主的汉字编码字符集　在此类汉字编码字符集中,中国、日本、韩国等使用汉字的国家和地区各自分别进行编码,它们的收字数目和位置排列各不相同。目前我国普遍使用的此类汉字编码字符集有如下三种。

(1)GB 2312—1980 标准(《信息交换用汉字编码字符集·基本集》)　这是我国颁布的第一个汉字编码字符集国家标准。其字符集由三部分组成,第一部分是字母、数字和各种符号,包括拉丁字母、俄文、日文假名、希腊字母、汉语拼音等共 682 个(统称为图形符号);第二部分为一级常用汉字,共 3755 个,按汉语拼音排列;第三部分为二级常用汉字,共 3008 个,按偏旁部首排列。

由于汉字与 ASCII 字符常常混合在一起使用,汉字代码如无特别标识,它与单字节的标准 ASCII 代码将混淆不清。为此采用的方法是用两个扩展 ASCII 码(两个字节)表示一个汉字,这两个字节的最高位都置“1”。这种高位为 1 的双字节(16 位)汉字编码通常称为 GB 2312 汉字的“机内码”,又称内码。它已经成为 GB 2312 汉字编码的一种事实上的标准。

(2)《汉字内码扩展规范》(GBK)　由于 GB 2312—1980 收录的汉字有限,不少人名、地名和古籍用字以及台湾、香港地区使用的繁体字并未收录在内,为此 1995 年我国颁布了 GB 2312—1980 汉字编码标准的扩展规范 GBK。GBK 规范中一共有 21 003 个汉字和 883 个图形符号,除了 GB 2312 中的全部汉字和符号之外,还收录了包括繁体字在内的其他大量汉字和符号,例如“計、機、國、節、圖”等繁体汉字和“冃、冄、冇、冇、鎔”等生僻汉字。随着 GB 18030 标准的推出,GBK 已经停止使用。

(3)台湾地区 CNS 11643—1992 汉字编码标准　其全称为《中文标准交换码》,共收入汉字和图形符号 4 万多个。与 CNS 11643—1992 对应的工业标准是 Big 5 码(俗称“大五码”),它是 CNS 11643 交换码的内码表示,通常用 Big 5 泛指二者。Big 5 共收录 13 000 多个汉字,在使用繁体汉字的地区(如台、港、澳门等地)广为采用。Big 5 码也是双字节代码,但它与 GB 2312、GBK 汉字代码不兼容,需要进行转换才能正确地进行显示与打印。

2. 包含汉字的多文种编码字符集　各个国家(地区)对以本地文字为主的字符集分别进行编码的做法产生了许多问题。例如,不同字符集的编码会互相冲突,同一代码可能代表多个不同的字符,或

不同的代码代表相同的字符；一台计算机需要支持许多不同的编码字符集才能进行多文种信息处理；数据在不同系统之间进行交换很不方便，等等。解决这些问题的途径是实现多文种的统一编码，即设计一种包含全世界所有文字符号的超字符集并进行统一编码。目前我国普遍使用的多文种编码字符集有下列三种。

（1）ISO/IEC 10646 标准——“通用字符集”（Universal Character Set，简称 UCS 标准）　这是国际标准化组织（ISO）和国际电工委员会（IEC）联合制定的多文种字符集标准。它包含了已知语言的几乎所有字符。除了拉丁语、希腊语、斯拉夫语、希伯来语、阿拉伯语、亚美尼亚语、格鲁吉亚语等之外，还包括中日韩统一汉字（CJK 汉字）。此外，UCS 还包含大量图形、印刷符号、数学符号和科学符号。

UCS 中的汉字是按照一定的规则将中国、日本、韩国等不同国家和地区使用的汉字进行认同和甄别，不论其字音和字义有无区别，只要字形（抽象字形）相同，该汉字就只有一个代码，这样整合在一起的汉字集合称为“CJK 汉字”。UCS 收录的 CJK 汉字共 74 000 余字，涵盖了 GB 2312 和 GBK 之全部，《现代汉语通用字表》之全部，Big 5 之全部，中国台湾地区新、旧电报码之全部，日本的 JIS 汉字之全部，韩国 KSC 汉字之全部。后来，随着越南文的加入，CJK 统一汉字也称 CJKV 中日韩越统一汉字。

UCS 标准规定，所有字符和符号都使用四字节进行编码（记作 UCS-4）。这样做的优点是编码空间极大，可以安排 13 亿个字符，能容纳足够多的各种不同文字符号，充分满足世界上多种民族语言文字信息处理的要求。可是，四字节的字符编码太浪费存储空间了。比较实际的做法是，在 UCS 编码空间中，把第 1 和第 2 字节均为“0”的一个子空间（称为基本多文种平面 BMP）作为其子集来使用，其中的所有字符均采用双字节编码表示（记作 UCS-2）。2000 年发布的 UCS 国际标准的第 1 部分（ISO/IEC10646-1：2000）中，规定 BMP 中收录 49 194 个字符，它们包括：①欧洲及中东地区使用的拉丁字母、音节文字；②各种标点符号、数学符号、技术符号、几何形状、箭头及其他符号；③27 000 多个 CJK 汉字。

（2）Unicode 编码字符集　这是由 Xerox、Microsoft、IBM、Apple、Adobe 等公司联合制订的一种工业标准，也称为统一码、联合码或万国码。20 世纪 90 年代开始，该标准与国际标准 ISO/IEC10646 相互对齐、合二而一。此后，两者一直保持相互协调、同步发展。

Unicode 的最新版本是 6.0.0 版（2010.10），其字符集中包含了世界各国和地区当前使用的 90 套书写符号近 11 万个字符。

为了适应各种不同的计算平台，与已经使用的字符编码保持向下兼容，Unicode 在计算机中具体实现时可以采用几种不同的编码方案。最常用的有两种：一种是称为“UTF-8”的单字节可变长编码；另一种是称为“UTF-16”的双字节可变长编码。

采用 UTF-8 编码方案时，标准 ASCII 字符仍以单字节代码（00H ~ 7FH）表示，带变音符号的拉丁字母、标点符号、希腊字母、阿拉伯文等音节文字使用双字节表示，CJK 汉字则使用三个字节表示，其他一些极少使用的字符用四字节表示。这样，它既保持了与传统 ASCII 编码兼容，又实现了各种字符集的统一编码。UTF-8 编码是 Unicode 编码文本的交换标准。目前，它已经在 UNIX 和 Linux 类操作系统中广泛采用。因特网的许多应用如 Web 网页和电子邮件等都支持 UTF-8 编码的使用。

采用 UTF-16 编码方案时，ASCII 字符、标点符号、希腊字母、阿拉伯文和 CJK 汉字等均使用双字节编码，其他不常用字符则使用四字节编码。UTF-16 编码方案已经被 Microsoft Windows、Mac OS 等操作系统及 Java 和.NET 软件开发环境等采用，它们都使用 UTF-16 来表示和处理文本信息。以 Windows XP 为例，其用户界面、文件名等所使用的字符都是 UTF-16 编码，Office 文档内部处理时也采用 UTF-16 编码，只有在数据文件需要保存到外存储器时才转换成为用户默认的本地编码。

UTF-16 编码又分两种：大尾序（Big-Endian，简写为 UTF-16 BE）和小尾序（Little-Endian，简写为 UTF-16 LE）。目前 Windows 系统对于 UTF-16 编码默认使用的是 UTF-16 LE，Mac OS 使用的是 UTF-16 BE。

（3）GB 18030—2005 标准（《信息技术　中文编码字符集》）　UCS/Unicode 中的汉字及其编码与我国已使用多年的 GB 2312 和 GBK 并不兼容。为了既能与国际标准 ISO/IEC 10646（Unicode）接轨，又能保持与 GB 2312 和 GBK 汉字编码标准兼容，保护用户多年积累的中文信息资源，我国制定了 GB 18030—2005 汉字编码国家标准，并规定在我国计算机类信息产品中强制执行。

GB 18030—2005 标准能完全映射 UCS/Unicode

字符集,并有所扩展。它采用单字节、双字节和四字节进行编码,码位总数达 160 多万个。它包含所有 CJK 汉字,同时还收录了藏文、蒙文、维吾尔文等主要的少数民族文字,能很好解决我国出版、邮政、户政、金融、地理信息系统等领域的用字问题。GB 18030 代码空间的分配及码位数目如表 1 所示。

表 1 GB 18030—2005 码位范围的分配

字节数	码位空间				码位数
单字节	0x00～0x7F				128
双字节	第 1 字节		第 2 字节		23 940
	0x81～0xFE		0x40～0x7E,0x80～0xFE		
四字节	第 1 字节	第 2 字节	第 3 字节	第 4 字节	1 587 600
	0x81～0xFE	0x30～0x39	0x81～0xFE	0x30～0x39	

GB 18030—2005 标准中,单字节代码共 128 个,与 7 位的标准 ASCII 字符的编码完全一致。双字节代码(23 940 个)用于表示 CJK 汉字,与 GBK 和 GB 2312 保持兼容;还有约 158 万个四字节编码则用于表示 UCS/Unicode 中的其他汉字和字符。

GB 18030 目前已编码的字符约 7.6 万个。随着我国汉字整理和编码研究工作的不断深入以及国际标准 ISO/IEC 10646 的发展,GB 18030 所收录的字符还会在新版本中逐步增加。

参考文献

1. 国家标准 GB 18030—2005(信息技术 中文编码字符集). 北京:中国标准出版社,2006

2. ISO/IEC 10646-1:Universal multi-octet character set-UCS-Part 1:Architecture and Basic Multilingual Plane

3. Unicode 官方网站:http://www.unicode.org/

(张福炎)

hanzi jianpan shuru

汉字键盘输入(Chinese character input via keyboard) 操作者通过键盘向计算机等信息设备键入**汉字**的过程、技术和方法,又称为汉字编码输入。它是计算机以及其他信息设备输入汉字的主要方法之一。

汉字是大字符集,专用的一键一字的汉字输入键盘由于键多、查找不便、成本高等原因早已不用,而利用只有几十个键的计算机键盘(甚至只有十几个键的手机按键)输入汉字时,无法使每个汉字与键盘上的按键一一对应,因此必须用一个或几个按键的组合来表示汉字,这就是汉字的键盘输入编码。

设计一种汉字键盘输入编码方案,首先要利用汉字的音、形等特征信息,按照一定规则,对指定的**汉字编码字符集**中的每一个汉字进行描述,然后再确定这些特征信息与键盘按键之间的对应关系。

汉字的键盘输入编码方案至少有几百种,从编码特征的角度看大体可以分成 4 类:①数字编码 这是使用一串数字来表示汉字的编码方法,例如电报码、区位码等,它们难以记忆,不易推广。②字音编码 这是一种基于汉语拼音的编码方法,简单易学;缺点是同音字引起的重码多,需增加选择操作。采用字音编码的汉字输入法叫做汉语拼音输入法,有全拼和双拼之分,全拼是按音节的字母来击键输入,双拼是按音节的声母和韵母来击键输入,声、韵各一键。③字形编码 这是将汉字的字形分解归类而给出的编码方法,重码少、输入速度较快,但编码规则不易掌握。采用字形编码的输入法主要有汉字部件(字根)输入法和汉字笔画输入法。前者选取若干有代表性的汉字的部件(又称字根)作为码元,以汉字字形结构及其切分规则作为编码的主要依据。后者采用选定的汉字基本笔画作为码元,其编码规则的制定需要先进行笔画认同,再规定编码的顺序。由于其码元少,可以在通用键盘数字键或专用小键盘上实现单手操作录入汉字。随着移动计算设备的普及,这种输入法正显示出其键元少的优越性。④音和形结合的音形码或形音码 它吸取了字音编码和字形编码的优点,使编码规则适当简化,重码减少,但掌握起来也不容易。

在上述编码输入方法的基础上,利用计算机的高速处理和存储能力,充分发挥计算机的统计学习功能,实现字词联想、词语联想,并采用词性、词法、词语搭配频率、句法甚至部分语义和语用知识来输入汉字,同时还自动记忆新词,自动调整词语优选顺序等,这些所谓的"智能汉字输入法",受到了广大用户的欢迎。

参考文献

1. 余锦凤,萧志春. 中文信息处理基础教程. 北京:北京大学出版社,2002

2. 李宝安,李燕,孟庆昌. 中文信息处理技术. 北京:清华大学出版社,2005

（杨尔虹）

hanzi shibie

汉字识别(Chinese character recognition) 利用计算机将书写或印刷的汉字图像自动转换为表示该汉字的计算机代码(例如汉字国标码、Unicode码等,参见**汉字编码字符集**)的过程、方法和技术。印刷体汉字识别可以自动高速地将大批印刷文字材料输入计算机,在数字图书馆、电子出版、数据库建设等领域有重要的应用。联机手写汉字识别是便携式信息设备输入汉字的主要手段之一。

汉字识别属于**模式识别**,它涉及图像处理、**人工智能**、统计决策理论、信息论、**中文信息处理**等学科,也涉及视觉心理学、语言文字学等。汉字识别过程包括汉字图像信号获取(印刷或手写书面文字利用扫描仪的扫描图像输入,或摄像机的图像摄影输入;联机手写汉字利用书写板传感器的书写笔迹获取,或触摸屏的触摸笔迹输入)、汉字行和字的分割、汉字特征提取和选择、基于汉字特征的汉字分类判决,以及利用上下文对分类判决结果的验证处理等。由于汉字的数量极大(数千乃至数万)、结构复杂、字形千变万化,在相当长的时间内,汉字识别是最困难的文字识别问题,也是最困难的模式识别问题之一。

光学字符识别(optical character recognition, OCR)最早起源于1929年,Taushek在德国获得了一项有关OCR的专利。欧美国家从20世纪50年代就开始了西文OCR技术的研究,以便代替人工键盘输入浩如烟海的文字材料。汉字识别的研究是1966年开始的。由于方块汉字完全不同于英文文字,汉字识别的研究受到人们的特殊重视,“汉字识别”也因此成为重要的技术名词。

1966年IBM公司的Casey和Nagy发表了第一篇关于印刷体汉字识别的论文。20世纪70年代以来,日本学者做了许多工作,其中有代表性的系统是1977年东芝综合研究所研制的可以识别2000个汉字的单体印刷汉字识别系统;80年代中期,东芝、松下、理光和富士等公司的基于专用硬件的印刷体日文识别系统走向市场。我国自70年代末开始了对汉字识别的研究,到80年代末研究开发成功印刷汉字识别系统和联机手写汉字识别系统的初级产品。在90年代,印刷体汉字识别和联机手写汉字识别的实用化产品逐渐完善,90年代末开始走向成熟,汉字识别已在国民经济的许多领域得到广泛应用。

汉字识别的任务是辨识印刷或手写生成的由来自成千上万种汉字字符组成的文本图像中的字符类别。由于字符的类别数成千上万,所以汉字识别属于超多类模式识别问题。汉字识别按照原始汉字图像信号输入计算机的方式的不同,分为脱机(输入)汉字识别和联机(输入)汉字识别两种;按汉字图像信号产生方式的不同,又有印刷体汉字识别和手写汉字识别两种。由此,汉字识别基本上可分为**印刷体汉字识别**、**联机手写汉字识别**以及**脱机手写汉字识别**3种。其基本流程如图1所示。

对于汉字文档的识别,除了解决单个字符识别以外,还必须解决汉字文档中单个汉字正确分割提取出来进行识别的问题。这包括文档版面分析、文本行切分、字符切分等问题。由于实际文档的多样性,特别是手写文档的多变性,使得字符切分成为汉字识别之后又一重要的困难问题。

对于汉字文档识别,不仅需要解决好字符串的字符切分和孤立字符识别外,由于文本上下文的语义相关,使得我们可以利用文本上下文的语义相关约束关系,将字符孤立识别转化为文本的语义相关识别,利用汉字识别后处理,可以极大地提高文本识

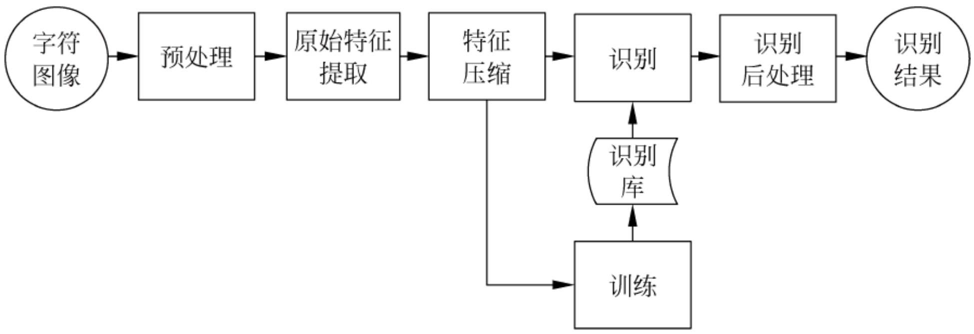


图1　汉字识别的基本流程

别性能。

经过 20 多年的努力，我国目前在印刷汉字识别、联机手写汉字识别和规则书写的脱机手写汉字识别的研究和实用系统的研发方面，均已取得重大进展，已经开发出高性能的实用系统，并广泛应用于各种领域。包括最困难的自由书写汉字识别也已取得了有限程度使用的进展。对于自由书写手写文档的识别，由于手写文档字符切分和识别格外困难，以及最近发展的摄像景物文字识别，还需要进一步研究和开发，实际推广应用尚需继续努力。

汉字识别基本方法

汉字识别的实现是利用模式识别或模式分类的原理和方法，对输入计算机的汉字图像信息提取识别特征，进行汉字类别的分类判定，以表达为代表该字符类别的计算机汉字编码。

汉字识别的方法基本上可以分为结构识别和统计识别两种。结构识别方法是在提取汉字笔画的基本结构基元的基础上，对汉字结构进行描述和匹配而识别的。但是，由于提取有限汉字结构笔画基元的不稳定性，结构识别方法对解决实际汉字识别问题成效不高。而基于汉字图像结构信息提取的汉字图像的高维统计特征，按照统计模式识别方法对汉字分类识别，由于高维统计特征具有汉字识别足够的互信息，以及统计模式识别的强鲁棒性特点，使得即使在实用中汉字受到强烈干扰和字形发生巨大变化的条件下，也能较好解决超大字符集印刷和手写汉字的识别问题。

汉字的笔画信息是汉字类别的本质特征。在统计识别中利用基于笔画的“微结构”形成的汉字字符图像总体笔画结构统计量作为汉字识别的特征，是一种比直接抽取笔画结构更鲁棒和更有效的利用笔画结构信息的方法。

在统计模式识别中，将提取的汉字样本特征用 N 维特征向量 $X=(x_1,x_2,\cdots,x_N)^{\mathrm T}$ 表示。汉字字符的模式类别数集合为 $\Omega=\{\omega_1,\omega_2,\cdots,\omega_C\}$，类别数为 C。在学习阶段，利用已知类别的样本进行学习，获得字符分类的鉴别函数 $g_i(X),i=1,2,\cdots,C$；在识别阶段，则利用鉴别函数 $g_i(X)$ 对未知类别的待识样本 X 的类别属性进行判决。即 $\omega(X)=\omega_i\in\Omega$ 当 $g_i(X)>g_j(X)$ 对所有的 $j\neq i,j=1,2,\cdots,C$。如果在学习阶段利用训练样本获得特征的类概率分布 $P(X|\omega_i),i=1,2,\cdots,C$，识别时按照贝叶斯最大后验概率准则进行判决，即 $\omega(X)=\omega_i\in\Omega$ 满足 $\omega_i=\arg\max_i P(\omega_i|X)$。将获得最小错误概率的分类，这是设计最优分类器的目标。

在简单近似的情况下，利用学习样本获得该类模式特征的平均向量 M_i，对于待识样本 X，利用最小距离判决函数，即 $g_i(X)=\|X_i-M_i\|^2$，获得最小距离分类器判决，即 $\omega(X)=\omega_i\in\Omega$，满足 $\omega_i=\arg\min\|X_i-M_i\|^2$。最小距离分类器识别方法简单易行，在某些情况下可以获得相当好的识别性能，从而得到比较普遍的使用。

一般情况下，印刷或手写汉字的特征的类概率密度分布可以用高斯分布来较好地近似，通过样本训练估计类条件概率密度分布的均值向量和类协方差矩阵，就可以获得贝叶斯最小错误概率二阶分类器 QDF，而最小距离分类器是在各类协方差矩阵均为么阵时的近似。由于高维二阶分类器训练时有大量参数需要估计，在样本数目不足的情况下会产生的维数危机问题，研究者提出的修正二阶分类器 MQCF，在困难的印刷和手写汉字识别中取得了成功的使用。

$$p(X|\omega_i)=\frac{1}{(2\pi)^{\frac{N}{2}}|\Sigma_i|^{\frac{1}{2}}}\exp\left\{-\frac{1}{2}(X-M_i)^{\mathrm T}\Sigma_i^{-1}(X-M_i)\right\}$$

在对整个汉字集合的识别系统中，对于每个汉字类别的鉴别函数的参数数据集合构成了汉字识别的所谓“识别字典”。例如，在应用最小距离分类系统时，“识别字典”中包括的是被识别汉字集合中每一汉字类别的特征向量均值的总和，在利用 MQDF 修正二阶分类器时还包括各类协方差矩阵的参数。

汉字识别的特征

为了识别汉字，必须要利用对汉字图像进行分析计算得到的，对该汉字字形结构关键性的、稳定的某种表征，作为汉字识别的特征，以便对汉字类别进行判决。模式识别信息熵理论证明，汉字识别特征的选择将最终决定汉字识别的性能。

汉字识别中，除了分类判决的鉴别函数外（参见“汉字识别基本方法”），最重要的是汉字的特征抽取和选择。应当说，特征选择决定了识别分类系统的鉴别性能，而鉴别实施的性能则由分类系统决定。识别特征的鉴别性能决定于特征所能提供的有效互信息熵，以最大互信息为准则进行特征的选择，可以较好满足识别的要求。在特征与识别模式间的有效互信息熵 I 表示为

$$I(F,E)=-\sum_{i=1}^{C}\int_{R^N}p(X,\omega_i)\log_2\frac{p(X|\omega_i)}{p(X)}\mathrm{d}X$$

式中：E 为模式类别空间，C 为类别数；F 为特征空间，X 为特征向量。$P(X,\omega_i)$ 为联合概率密度函数。

有效互信息熵准则说明，一个好的识别特征必须满足：①特征与识别类别密切相关；②特征对类内变化比较稳定，受到噪声和字形变化影响较小。因此，汉字识别的研究过程，也包括对特征选择的不断发展，以及对汉字图像识别的特征选择问题的进一步认识。

汉字识别的特征有粗外围特征、粗网格特征、复杂指数和四边码、笔画密度特征、汉字特征点、边框和局部特征、部件模板、笔画方向和轮廓特征、网格单元、笔画序列和各种数学变换特征，等等。这些特征在识别汉字时各有特色，互有优劣，曾被应用于各种识别方案的粗、细分类中。汉字识别研究发展至今，人们认识到汉字识别特征抽取的方法可以有多种，其中基于汉字的整体图像获取充分的信息最重要，例如，基于梯度的方向线素的网格特征取得了比较优秀的识别效果。

为了获得优良鉴别能力的识别特征，在原始抽取的特征上进行特征变换，也成为提高汉字识别性能的有效手段。例如，对原始特征向量进行主分量分析（principle component analysis，PCA），进行维数压缩，减少学习样本不足产生的维数危机问题；利用线性鉴别分析（linear discriminate analysis，LDA），获取最有鉴别能力的特征分量等。这些利用特征变换使特征鉴别能力优化的方法，都将在一定特征维数压缩的条件下有效地提高和改进汉字识别系统的识别能力。

汉字识别后处理

利用语言模型（句法和语义规则）描述的文本上下文关系，对汉字文档图像识别中汉字识别的结果加以约束，减少识别结果的不确定性，对提高文档识别性能十分重要。

文档记录书面语言，其内容受到上下文语义的限制。利用这种上下文语义约束就可以对文本图像识别的结果进行改进。未知汉字图像由汉字识别分类系统获得的识别结果是一个按一定置信度递减顺序排列的候选字符串（参见汉字识别基本方法）。在不考虑上下文语言模型的情况下，是将识别结果的候选字符串中最小距离（或最大可信度）者，作为最终的识别结果。对于未知汉字文档图像的识别，可以在考虑上下文约束的情况下，从识别候选字符串中选择满足整个词或句子与语言模型相匹配的最优结果作为最终识别的结果。这种考虑上下文约束的识别后处理方法基本上是基于语料库的统计语言模型的方法的。它是将基于统计的马尔可夫语言模型应用于文字识别。假定语言是一个 $n-1$ 阶的马尔可夫链，称这种语言模型为 n 元文法模型（n-gram 语言模型）。利用文本的大规模语料库进行加工、统计，可以获得文本语料 n 元文法模型的同现概率和条件概率等参数等。

在利用 n-gram 语言模型进行上下文后处理时，用动态规划方法从单字识别给出的识别候选字符序列中找出构成出现概率最大的一个句子，从而在一定程度上纠正单字识别时存在的错误。

n-gram 语言模型，可以是基于字为单元的，也可以是基于词为单元的，还有基于词类的 n-gram 等。常用的字 bigram 模型是利用相邻两个字间的约束关系的，由于常用汉字的数目一定，它处理速度快而且易于实用。基于词的 n-gram 模型是利用能表示独立词义的相邻词间的约束关系，但必须要对文本进行分词处理，而数万以上词的数量远超字的数量，给模型的训练带来巨大困难。

利用语料库统计模型的上下文处理方法可以自动纠正句子中连续几个字的错误，对于识别性能的提高有较明显的作用，在文档识别中得到了广泛应用。

印刷文本版面分析

印刷文档均具有一定的排版格式，对文档的识别和理解首先需要对印刷文本版面的排版格式进行自动分析、切分和标识。

由于书面文字信息均按照一定的版面排版印刷成为文本图像，它包含了图、文、表格等各式文档信息，印刷文本记载和表述的是由字符串有机组成的文字信息，书本、杂志、报纸等数字化时得到的是整个页面的图像。对于印刷文本的识别，首先需要从整个页面图像中将文字块分割出来。而要将文字块从复杂的版面中分割出来的过程，需要进行版面分析。即经过版面分析，将文档页面的各种图文属性内容进行分区分割，再将文字块从版面中分割出来。然后，对文字块图像进行文字行切分和文字切分的逐步处理，最后获得单个字符的图像，再针对单个字符图像进行字符识别，获得单字字符的字符编码表示的识别结果。汇总单字识别结果，得出整个篇章和文档的数字化结果。因此，除了要求正确的单个字符识别以外，印刷文本的版面分析和文本图像中字符图像的正确切分都是文档（文本）正确识别的前提。

印刷文本是具有一定版面排版格式的印刷品。印刷文本的自动版面分析是对印刷文本版面的排版自动进行区域分割、属性判决和标识的过程。实际的印刷文本是由若干不同属性的区域，如文本、图像、图形、表格等组成。文本区域则包括标题、作者、正文、公式、注释、页码等。正文块又可分为单栏、双栏、三栏等编排格式。文本块之间用空白条、直线、装饰线、花边等隔开。这些版面不同区域的分割和属性的标注，以及复杂版面中文本块间的阅读顺序和篇章分割等，都是版面自动分析和理解的重要内容，它是文本识别的重要内容之一，更是文档全信息数字化和文档自动索引和标注生成的重要手段。

由于版面的复杂和多变性，复杂版面(如报纸版面，见图2)的自动版面分析和理解是一个十分困难的问题。一般有按自上而下(主要利用投影方法)和自下而上(逐级合并方法)两种算法。近来又按照利用形状分析和纹理分析将算法分为两类。有多种新算法提出，如基于形状的方法有多层次置信度指导下的自底向上的版面分析算法，基于纹理的算法有统一的隐马尔可夫模型(HMM)版面分析框架及


人民日报
RENMIN RIBAO
天津精心组织文化精品生产
在"五个一工程"评选中连续四年"满堂红"
讲文明、树新风，先进典型学习贯彻十五大精神
推动精神文明建设再上新台阶
制定优惠政策 广开就业门路
湖北再就业率连续三年超过80%
早动手早得益
没有后顾之忧了
十五大报告是及时雨
十五大精神在基层⑦

图2 复杂的报纸版面分析

算法等。图3给出了一个文档识别系统的流程图。

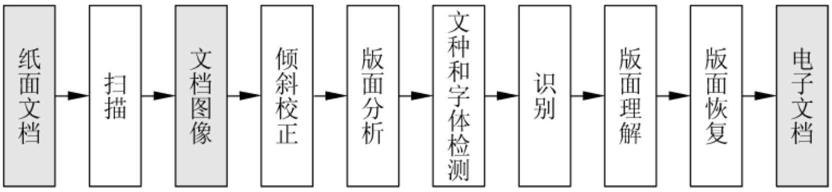


图3 包括版面分析和复原的文档识别系统流程图

对于文档信息而言，除了字符表示的文字信息以外，版面排列、标题、栏目以及字符大小和字体等，也往往包含重要信息。保留整个版面信息的文档数字化称为全信息数字化，是文档数字化的重要内容。

参考文献

1. 郭平欣，张淞芝. 汉字信息处理技术. 北京：国防工业出版社，1985

2. 张炘中. 汉字识别技术. 北京：清华大学出版社，1992

3. 丁晓青，王言伟，等. 文字识别：原理、方法和实践. 北京：清华大学出版社，2016 （丁晓青）

hanzi xinxi chuli

汉字信息处理(Chinese character information processing) 利用计算机对**汉字**和汉字文本进行加工和操作的技术，如汉字字符集的确定、编码、输入、输出、识别、转换、字型表示与存储、字频统计、汉字属性库构造以及中文文本的编辑与排版等。

当代计算机内部最基本的信息单位是二进制字位(bit)，它只采用"0"和"1" 两个符号。计算机交换信息的单位一般是由8个字位组成的字节(byte)，每个字节可有256种不同编码。在英语国家，人们通常使用的字符系统只包含几十个字母、10个阿拉伯数字以及一些标点、符号，256种编码足以应付自如。计算机与英语文化的这种适应性为计算机在英语国家的迅速普及、广泛应用提供了方便。当计算机引入中国之后，汉字文化与这种先进的工具不相适应的矛盾便凸现出来了。汉字的数量远远超过了一个字节及其编码可以表示的范围。计算机内部至少要用2个字节长度的编码才能代表数以万计的汉字。仅这一个问题便给汉字在计算机的内部操作和计算机内部各部件之间以及计算机网络的通

信带来很多麻烦。其他如字型的表示与存储、汉字的输入与输出等也都遇到了一系列在处理英语时所碰不到的难题。

在汉字信息处理的诸多任务中,人们最关注的有两个问题:一是汉字的输入;二是汉字文本的排版和印刷。汉字输入分为键盘输入(参见**汉字键盘输入**)和非键盘输入两种。所谓键盘输入就是利用电脑键盘上的几十个字符键乃至手机上的10个数字键向机器输入汉字,为此必须对汉字进行"编码"。所谓"编码"就是用键盘上的若干个字母或数字代表一个汉字。在机器内预装汉字及其编码的对照表(码本)和转换软件。最容易使用的编码方案是采用汉语拼音,不过存在编码较长、同音字有"重码"、不知道读音则无法输入等缺点,因此基于字形的编码方案也有用武之地(参见汉字编码字符集)。非键盘的方法则采用文字识别(参见**汉字识别**)、言语识别(参见**汉语言语识别**)等技术。汉字文本的排版和印刷(参见**计算机辅助出版**)则要克服巨量汉字字形(参见**汉字**)信息的压缩存储、快速还原和高质量输出等技术难题,中国专家为此做出了卓越贡献。1987年《经济日报》采用北京大学研制的激光照排系统出版了世界上第一张采用计算机屏幕组版、整版输出的中文报纸,在中国掀起了"告别铅与火,迎来光与电"的印刷技术革命。其他像中文速录机也是很成功的汉字信息处理应用系统。

汉字信息处理是**中文信息处理**的基础阶段,是实现计算机中文化的前提。但汉字信息处理研究也并不局限于单个的"字",需要向**汉语信息处理**的方向发展。仍以输入为例,如果按词输入,即便使用拼音方案,重码率也可大幅度降低。现在以语句为单位的拼音输入技术已被广泛使用,这种技术就是建立在利用上下文信息的语言模型上的。又如,在**模式识别**之后进行基于语言学模型的后处理可以显著提高文字识别和语音识别的正确率。

参考文献

1. 郭平欣,张松芝. 汉字信息处理技术. 北京:国防工业出版社,1985

2. 余锦凤,萧志春. 中文信息处理基础教程. 北京:北京大学出版社,2002

(俞士汶)

hong chuli chengxu

宏处理程序(macroprocessor)　把源程序中的宏指令或宏语句扩展成等价的、预先定义的指令序列或语句序列的处理系统。

宏指令或宏语句实际上是按规定格式书写的某一源程序段的缩写。它们通常是用户根据自己的特定需要,采用程序设计语言所提供的指令或语句来定义,称之为*宏定义*,其中应给出宏的名字、格式、参数和等价的指令序列或语句序列。对于常用的宏指令或宏语句亦可由系统预先定义,供用户直接引用。当用户在程序中要使用宏指令或宏语句功能时,只要按宏定义的格式,给出宏的名字及其相应的参数即可,这称之为*宏调用*。当宏处理程序将源程序中出现的宏调用扩展成等价的宏指令序列或宏语句序列时,称之为*宏扩展*。建立宏处理程序后,用户可以方便地定义和使用自己所需的宏指令或宏语句。这不仅能简化应用程序的编写,而且有助于软件人员研究和移植有关的软件。例如,利用宏指令或宏语句设计虚拟机,研究新的语言,以及生成带有变化成分的软件等。

宏处理程序通常采用两遍算法实现:第一遍收集宏定义的信息;第二遍对源程序中的宏调用实施宏扩展。在第一遍扫描中,遇到宏定义时,应把名字、格式、参数等信息以及随后的等价的指令序列或语句序列记录到宏定义表中。对于源程序中宏定义以外的部分,将不加改变地复写到目标程序区中。第二遍扫描第一遍所产生的中间结果程序。遇到宏调用时,则将宏定义表中相应的等价指令序列或语句序列复写到目标程序区中。复写过程中,要用宏调用中的实在参数替换宏定义中的形式参数。如果限制每个宏调用只能调用前面已定义的宏指令或宏语句,那么这种宏处理程序的实现算法可合并成一遍算法来完成。

功能较强的宏处理程序还可增加嵌套宏定义、嵌套宏调用或条件宏处理等功能。如果宏定义中含有另外的宏定义,则称为嵌套宏定义。如果宏定义A中出现宏调用,那么在扩展A的宏调用过程中,又要进一步转去扩展其他的宏调用,这种情况称为嵌套宏调用。如果宏处理程序能根据宏调用中的特殊参数,有选择地把宏调用扩展成不同的指令序列或语句序列,则称为条件宏处理。

参考文献

1. Campbell-Kelly M. An introduction to macro. London: MacDonald, 1973

2. Brown P J. Macro processor and techniques for portable software. London: Wiley, 1974

3. Cole A J. Macro processor. Cambridge: Cambridge University Press, 1976

(曹东启)

hongmo shibie

虹膜识别(iris recognition)　计算机利用人的虹膜纹理信息自动识别其身份的过程和技术。虹膜是位于眼睛黑色瞳孔和白色巩膜之间的圆环状部分,呈现一种由内向外的放射状结构,包含有许多相互交错的类似于斑点、细丝、冠状、条纹、隐窝等形状的细微特征,如图1中圆环内部区域所示。以上这些细微特征称为虹膜的纹理信息。它们主要由胚胎的发育环境差异所决定,对每个人来说都是唯一的,因而可以用来进行个人身份的认证。测试表明,虹膜识别通过合适的算法其准确性是所有**生物特征识别**中最高的。

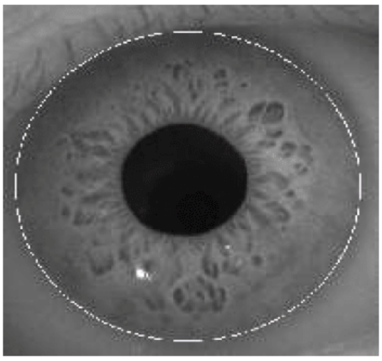


图1　虹膜图像(图片引自参考文献1)

虹膜识别的思想可以追溯到20世纪30年代。1936年,眼科专家Frank Burch指出虹膜具有独特的信息,可以用于身份的辨识,但是并没有引起广泛的注意。后来直至1987年才由医生Leonard Flom和Aran Safir提出了利用虹膜图像进行自动识别的概念并获得专利。1991年美国的Johnson实现了第一个自动虹膜识别系统。1993年John Daugman研制成功高性能的虹膜识别算法,得到大家的认可并仿效。其后虹膜识别的研究得到飞速发展。

虹膜识别系统在国外已经日趋成熟,并且已经在金融、公安、国防等领域得到初步应用。国内虹膜识别的研究也日趋成熟,中国科学院自动化研究所、清华大学、香港理工大学等院校和科研机构都开展了虹膜识别的相关研究,并取得了很大的研究进展。

虹膜识别主要包括虹膜的检测与定位、特征的提取以及识别等。虹膜检测从输入的原始图像中确定虹膜与巩膜的外边界以及虹膜与瞳孔的内边界,以提取其中的虹膜部分作为后续工作的基础。而特征的提取则从虹膜图像中选择所需要的特征用于识别。

特征的提取是虹膜识别的关键。目前国际上比较有影响的方法是Daugman于1993年提出的基于相位分析的方法,该方法采用Gabor小波滤波的方法编码虹膜的相位特征,然后利用归一化的Hamming距离实现特征匹配。其基础是Gabor小波具有和人类简单视觉细胞相似的视觉特征,可以很好地分析现实世界中的各种模式。基于相位分析的方法是目前识别性能最好的方法。1998年Boles等人提出了基于过零点检测的方法。该方法采用一维小波对沿虹膜中心同心圆的一条采样曲线进行虹膜中纹理的过零点检测,将其作为虹膜的特征,并进行识别分类。另外一类比较常用的方法是基于纹理分析的方法,比如可以采用Gabor滤波器提取虹膜在不同频率和方向下的纹理信息,或者采用Haar小波对虹膜图像进行分解提取其中的高频信息作为特征等。

2001年英国国家物理实验室的测试表明,虹膜识别通过合适的识别算法,其准确性是所有生物特征识别中最高的。因此在一些安全性能要求较高的领域,虹膜识别具有广阔的应用前景。目前虹膜识别的主要难点在于鲁棒的特征描述和提取方法。由于瞳孔随着光照条件的变化而引起虹膜的形变,眼球的旋转、睫毛对于有效虹膜区域的遮挡等,这些都会影响到特征提取算法的性能。虹膜识别的另外一个难点在于活体虹膜的检测,也是目前研究人员关注的焦点。

参考文献

1. Daugman J G. Biometric personal identification system based on iris analysis. United States Patend, No. 5291560, 1994

2. Wildes R P, Asmuth J C, et al. A machine-vision system for iris recognition. Machine Vision and Applications, 1996, 9: 1-8

3. Boles W W, Boashah B. A human identification technique using images of the iris and wavelet transform. IEEE Trans on Signal Processing, 1998, 46: 1185-1188

(蔡莲红　吴志勇)

hucaozuo xieyi

互操作协议(interoperation protocol)　在分布式计算环境下,位于网络层和传输层之上,规定软件实体之间进行互操作时的消息语法和语义、数据规范以及消息传递方式的协议。

互操作协议是软件实体(尤其是异构软件实体)之间进行信息交换和协同工作的基础。互操作

协议通常由消息协议、数据规范以及消息协议到传输协议的映射等几部分构成。消息协议规定交换过程中的各种消息类型、格式和语义;数据规范定义数据的表示、编码、格式与语义等;消息协议到传输协议的映射规定软件实体间信息交换的通信方式。

目前,最典型的互操作协议是 GIOP(General Inter-ORB Protocol)/IIOP(Internet Inter-ORB Protocol)协议。GIOP 协议是公共对象请求代理结构 CORBA 中对象请求代理 ORB 之间通用的互操作协议,它规定了对象请求代理之间所传递的消息类型、消息格式、相关的数据结构和数据类型及其表示方式。GIOP 协议本身独立于底层的传输层协议,可以被映射到任何一个面向链接的传输层协议。具体而言,GIOP 协议主要包括:①对象请求代理之间通信时所使用的消息类型及其格式和语义。消息类型主要包括请求、应答、取消请求、请求定位、应答定位、连接关闭、消息错误以及分段等,对象请求代理之间通过这些消息类型进行通信和交互。②采用公共数据表示方法 CDR(common data representation)来表示各种消息所携带的数据(请求参数和结果等)。公共数据表示方法通过建立统一的二进制表示方法,可以支持不同语言数据类型之间的互操作,例如该方法同时支持高字节和低字节表示,使得使用高字节序和低字节序表示数据的机器都可以按其格式发送数据,如果始发方和接收方使用不同的字节序,接收方负责字节序转换。③面向具体传输层协议的映射,GIOP 协议到 TCP/IP 协议的映射称为 IIOP(Internet Inter-ORB Protocol)协议,IIOP 协议是 GIOP 协议基于 TCP/IP 协议的一个具体实现,它是互联网内对象请求代理之间标准的互操作协议。

在基于 GIOP/IIOP 互操作协议的分布式应用中,当客户应用想访问服务器应用中服务对象中的一个方法时,首先将服务请求发送给客户方的对象请求代理机制,对象请求代理机制在接收到客户请求后,将客户请求按 GIOP 协议规定将其封装成一个 GIOP 请求消息,但并不对请求内容进行解析;而后,对象请求代理使用 IIOP 协议对请求消息进行处理,如地址信息的抽出、连接到建立等,并使用 IIOP 协议将经过处理的消息发送给服务器方,从而实现双方的互操作。

面向不同应用需求和不同层次的其他互操作协议主要有 SOAP、HTTP 等协议。HTTP(HyperText Transport Protocol)是一种应用层的面向对象的超文本传送协议,规定了浏览器和万维网服务器之间互相通信的规则,支持将超文本标记语言 HTML 文档从 Web 服务器传送到 Web 浏览器;SOAP(Simple Object Access Protocol)是一种轻量的、基于 XML 并能够与 HTTP 绑定的简单对象访问协议。

互操作协议作为分布式计算环境下实现软件实体之间的通信和交互基础,是软件实体之间协同、共同完成任务目标不可或缺的通信协议,在分布式应用中发挥着重要作用。随着应用需求的发展和分布计算环境异构性的不断增加,面向不同层次和应用需求的互操作协议也将得到进一步的扩展和发展。

参考文献

OMG. Common Object Request Broker Architecture Specification. http://www.omg.org/technology/documents/corba_spec_catalog.htm（史殿习　丁博）

hulian wangluo

互连网络(interconnection network,ICN)

通过直接或间接方式将一个并行计算机系统中的各个处理单元(也称为结点)或多个处理机、存储模块以及各种外部设备相互连接起来,在系统软件控制下,相互通信和协调工作的硬件网络结构,它是并行计算机系统重要的组成部分。

并行计算机系统内部的互连网络与连接多个计算机系统的局域网或广域网在地理分布、传输速度与控制方式等方面有区别。因此,互连网络这一术语一般只在讨论并行计算机内部通信时使用。但是,随着通信技术和分布式计算技术的进步,过去用于远程通信的装置也逐渐用来作为松散耦合并行计算机系统或机群系统的通信部件,互连网络与一般计算机网络的界限将越来越模糊。

在并行计算机出现以前,电话通信已经广泛使用。20 世纪五六十年代电话通信事业的迅速发展,推动了开关连接系统的基本理论研究,其成果集中反映在 C Clos 关于非阻塞开关网络的论文和 V Benes 关于连接网络和电话通信数学理论的经典著作中。并行计算机互连网络中非阻塞 Clos 网和可重构 Benes 网即来源于这些成果。互连网络中常用的交叉开关也是采用电话通信中的术语。互连网络最先在美国伊利诺依大学研制的 ILLIAC Ⅳ阵列机中采用,这种网络当时主要用于处理机单元之间的数据置换,故当时称为置换网络。美国卡内基-梅隆大学于 1974

年研制成功的包括 16 台处理机的多处理机系统 Cmmp 最早采用交叉开关实现存储器共享。美国 BBN 公司 1989 年推出的 TC-2000 率先在商品化并行计算机中采用多级互连网。Intel 公司和 NCUBE 公司在其推出的大规模并行计算机中分别采用的网格与超立方体互联网是点到点静态互联网的代表。

分类与结构参数

互连网络可分为静态与动态两大类。在静态互连网络中，作为通信用的开关元件和缓冲寄存器等分散设置在每一个结点内，各结点之间的连接在设计机器时已经固定，程序运行时的连接关系不能动态设置。静态互连网络中结点间的连线是无源的，其连接通路称为被动连接通路。这种互连网络应和应用问题的通信模式匹配，不同的静态互连结构适合于不同的应用。静态互连网络的主要优点是可伸缩性强，成本较低。已生产的大规模并行计算机普遍采用这种互连网络。与此相反，动态互连网络通过控制信号控制网络中的开关元件与仲裁部件，按程序运行的要求建立系统中各功能部件之间的连接通路。因此，处理机之间或各功能部件之间的连接可动态改变。这种互联网具有较强的通用性，对应用问题的通信模式限制较少。但相对于静态互连网络，其成本较高，技术较复杂，可伸缩性较差。大多数通用的多处理器系统都采用动态互连网络。

描述一个互连网络通常采用结点度、网络直径、对分宽度以及对称性等参数。在表示网络拓扑结构的图中，结点所连的边数称为该结点的度。结点度反映每个结点的输入输出接口数，因而直接与网络成本有关。网络中任意两结点间沿最短路径通信所经过的边数称为两结点的距离，网络中任意两结点间距离的最大值称为网络的直径。网络直径与通信延迟有一定的关系。如果将一互连网络分成相等的两半，在各种对分方法中，连接这两半的最小连接边数称为网络的对分宽度。一个好的互连网络应当通信能力强、成本低而且可伸缩性好，这就要求结点度小而且一致、网络直径小而且随结点数增加只缓慢增加、对分宽度适中而且网络对称性好、不含通信瓶颈。

静态互连网络的拓扑结构　静态互连网络可以有各种不同的拓扑结构，较常见的包括线形网、环形网、带弦环形网、树形网、星形网、网格网、超立方体网和全连接网等。一维的线形网最简单，但网络直径最长，从一端向另一端发送信息要经过网中所有结点。环形网将线形网两端连接起来，所以网络直径减少一半。如果将环形网上各结点的度从 2 增加到 3 或更大，就构成带弦环形网。在结点度为 3 的静态互联网中，二叉树是较理想的互连方式。树形网的直径比环形网更小，只有 $2\log_2 N$（N 是结点数）。但此网络中没有冗余通路，容错性能较差，而且系统中信息流量不均匀，根结点附近的结点可能成为通信瓶颈。这一缺点在胖树网中得以克服。在胖树网中，从叶结点到根结点，通信带宽逐步增加。星形网是一种两层的高结点度树形网。网格网是将邻近结点按规则的平面几何图形连接起来的互连网络。矩形连接的网格网是最常用的网络结构之一。早期的并行计算机 ILLIAC Ⅳ，DAP 和后来的大规模并行计算机 CM-2，Paragon 都采用网格网。纯粹的网格网是不对称的，边界结点的度数少于内部结点。为了减少网络直径，可将网络周围的结点对接起来形成筒形、网状环等变形的网格网。

更复杂的拓扑结构包括**超立方体网**、k 元 n 立方体网及其变形结构。一个 n 维超立方体网由 $N=2^n$ 个结点组成，其直径等于维数，结点之间的距离小于网格网与环形网，但这种结构的高维网扩展性较差，每次扩展要增加 1 倍结点。k 元 n 立方体网与超立方体网的区别是它的每一维包含 k 个结点，而超立方体网的每一维只有两个结点。**立方体连结环**（CCC）是将 k 维超立方体的每一个结点用一个含 k 个结点的环代替而形成的网，它的主要优点是每一结点的度都是 3，使网络易于扩展。高维网络每一维有专用的通信链路，不与其他维共享，因而通道的利用率低于低维网络。业已证明，对于给定对分带宽的互连网络，低维网络比高维网络延迟低、冲突少、吞吐量大。20 世纪 90 年代推出的大规模并行机大多采用低维互连网络。图 1 给出了各种静态互连网络的拓扑结构，表 1 比较了各种静态互连网络的特征。

动态互联网　动态互连网络包括总线、多级互连网络和交叉开关网络，已运用于单指令[流]多数据流和多指令[流]多数据流并行计算机。

总线由一组连线及控制器件组成，供处理机、存储器模块以及外部设备传送数据之用。总线基于分时方式工作，每一时刻只允许与它相连的一对设备进行通信，当有多个通信要求同时出现时，总线仲裁器要按优先级进行仲裁。设计总线的关键技术包括总线仲裁、中断控制、一致性协议和传输进程。根据系统中各部件传输数据的不同速度，总线往往分成

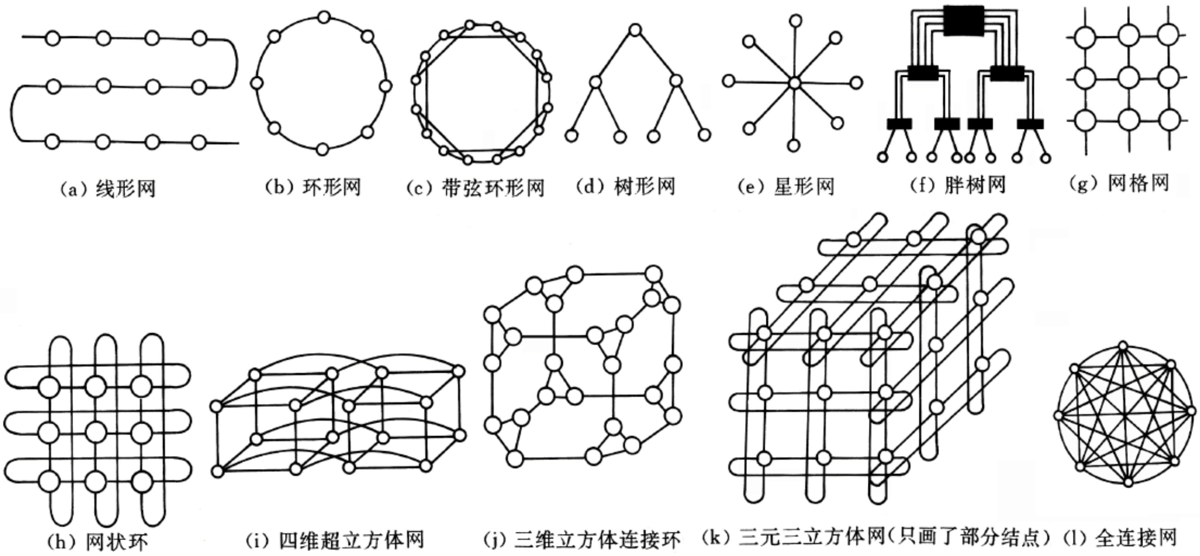


图 1 静态互连网络拓扑结构

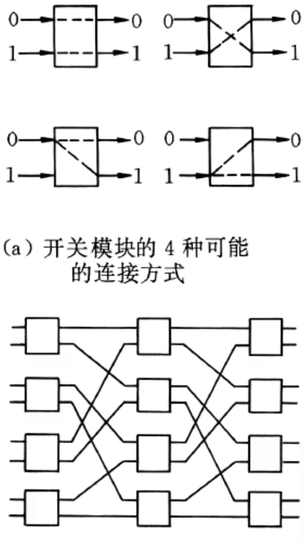
表 1 静态互连网络的特征

网络类型	结点度	网络直径	链接数	对分带宽	是否对称	备 注
线形网	2	$N-1$	$N-1$	1	否	N 为结点数
环形网	2	$\lfloor N/2 \rfloor$	N	2	是	N 为结点数
二叉树网	3	$2(h-1)$	$N-1$	1	否	$h=\lceil \log_2 N \rceil$
二维网格网	4	$2(r-1)$	$2N-2r$	r	否	$r=\sqrt{N}$
二维网状环	4	$2\lfloor r/2 \rfloor$	$2N$	$2r$	是	$r=\sqrt{N}$
超立方体	n	n	$nN/2$	$N/2$	是	$n=\log_2 N$
CCC	3	$2k-1+\lfloor k/2 \rfloor$	$3N/2$	$N/(2k)$	是	$N=k\times 2^k \quad k\geqslant 3$
k 元 n 立方体网	$2n$	$n\lfloor k/2 \rfloor$	nN	$2k^{n-1}$	是	$N=k^n$ 结点
全连接网	$N-1$	1	$N(N-1)/2$	$(N/2)^2$	是	N 为结点数

几个层次,直接与多处理机和存储器相连的总线速度最快,一般称为局部总线或内部总线;与外部设备相连的总线称为外部总线或系统总线(参见**系统总线**)。为了便于多个功能部件通过印制板互连,多处理机的系统总线往往在计算机的底板上实现。随着技术的进步,总线的通信能力不断提高,商品化多处理机的内部总线带宽已超过 1 GB/s。

多级互连网络由多级开关模块和级间连接组成。一个 $a\times b$ 开关模块具有 a 个输入端和 b 个输出端,通常采用的开关模块规模为 2×2,4×4,8×8。开关可以动态设置以建立合适的输入输出连接。开关模块允许“一对一”或“一对多”的输入输出映射,但不允许“多对一”的映射,即必须避免输出端的冲突。当只允许一对一映射时,开关模块实际上是一个$n\times n$交叉开关。较常采用的级间连接方式包括全混洗、交叉开关和立方体连接等。一种较流行的多级互连网络叫 Omega 互连网络,它采用2×2 的开关模块和全混洗方式的级间连接。图 2 表示 8×8 Omega网中 4 种可能的开关连接方式与级间互连模式。一般来讲,N 个输入的 Omega 网需要 $\log_2 N$ 级 2×2 开关模块,每一级需要 $N/2$ 个开关模块,一共需$(N\log_2 N)/2$个开关模块,每个开关模块单独进行控制。IBM 公司的大规模并行计算机 SP-2 是采用多级互联网的代表。

交叉开关是连接能力最强的动态互连网络。它通过一个 $n\times n$ 开关矩阵动态地实现 n 个输入到 n



(a) 开关模块的 4 种可能的连接方式

(b) 全混洗级间连接

图 2　8×8 Omega 互连网络

个输出的任意置换。根据运行程序的要求,每一开关单元可以动态设定为开或关状态。交叉开关既可实现处理机与存储器模块之间的互连,又可以实现处理机之间的互连。例如日本富士通公司于 20 世纪 90 年代推出的 VPP 500 向量并行计算机采用 224×224 交叉开关连接处理机。处理机与存储器模块之间的交叉开关可使一个处理机同时发出对不同存储器模块的请求,实现并行存取。但对处理机之间的交叉开关,同一时刻交叉开关的每一行和每一列都只允许最多 1 个开关打开。交叉开关的行类似多总线。当多个处理机同时要求存取某一存储模块或与某一处理机通信时,交叉开关必须有某种排队或仲裁机制处理冲突。

在上述 3 种动态互连网络中,总线的成本最低,但每个处理机可得到的通信带宽较低,扩展性较差,因此只有规模较小的多处理机系统(一般几个到几十个处理机)采用总线互连。交叉开关是通信能力最强但成本也最高的互连网络。它的硬件复杂性与处理机数量的平方成正比。当并行计算机系统中的处理机数量很大时,采用交叉开关互连成本太高。多级互连网络的通信带宽与成本都介于总线与交叉开关之间,其优点是具有较好的模块性与可扩展性,但其通信延迟与开关的级数成正比,对于 N 个处理机,则有 $\log_2 N$ 的通信延迟,而总线与交叉开关的延迟原则上都与处理机个数无关。

互连网络的工作一直围绕增强网络的连接能力与降低成本两个方面进行。所谓连接能力是指网络能同时连通的最大连接数。由于竞争通信线路,有些通信请求不能同时满足,这种状态称为阻塞或冲突。上面介绍的多级互连网络大都是阻塞型网络。一般,开关交叉点越多,连线越多,成本就越高,但建立连接的路径也越多,阻塞的机会越少,连接能力也就越强。交叉开关有 n^2 个开关单元,它是典型的非阻塞网络。有些互连网络通过重构,即重新设置各开关模块的状态,也可以消除冲突。这种网络称为可重构非阻塞网络,但这种网络的开关控制算法相当复杂。当互连网络的规模日益增大时,容错性成为另一个重要的要求,提供冗余的连接通路是实现网络容错的必要条件。

发展趋势

20 世纪 90 年代以来,采用工业上大量生产的主流处理机芯片或市场上流行的工作站构成并行计算机系统已成为发展高性能计算机的主要途径,因此处理机互连技术已成为区别不同并行计算机系统的主要标志。并行计算机系统互连水平的高低不仅体现在互连网络的硬件水平上,更主要地反映在支持处理机之间的通信和协作的系统软件上,特别是路由选择、流控制、容错等有效算法的实现。互连网络的设计必须考虑处理机的内部功能,尤其是互连网络与处理机的接口。

有关互连网络的各种工业标准已经建立或正在建立。电气和电子工程师学会(IEEE)已经通过**可伸缩一致性接口**(SCI)标准。这是一种基于点到点通信的高速总线标准,为研制、生产高性能的多处理机系统提供了一条途径。**异步传送模式**是下一代多媒体通信的工业标准,为语音通信和数字通信提供了高速有效的协议。未来的计算机互连网络的一种方式可能会建立在异步传送模式基础上。

光通信技术发展迅速,光互连可以提供很高的通信带宽,而且噪声低、功耗小。但光开关器件的速度与集成度还赶不上电子器件,因此,在可以预见的未来,路由选择和通道管理等开关器件仍将主要采用电子器件。电子开关器件与光通信线路的集成将是今后计算机互连网络的主要发展方向。较小规模的多处理机系统主要采用电子线路互连,光互连网络将首先在较大规模的并行机与远距离分布式系统中得到应用,并将逐渐成熟而成为互连网络的主流。

参考文献

1. Siegel H J. Interconnection networks for large-scale parallel processing: theory and case studies. New York: McGraw-Hill, 1989

2. Hwang K. Advanced computer architecture. New York: McGraw-Hill, 1993
（李国杰）

hulianwang

互联网(Internet)　全球最大的、开放式的、由众多网络互联而成的计算机互联网。这互联网的一般性定义，意味着全世界采用开放性协议的计算机都能互相通信。狭义的互联网指上述网中所有采用IP协议(参见**网际协议**)的网络互联而成的网络，通常把这样一个网称为IP网。在IP网中，TCP/IP协议的分组可通过路由选择相互传送。

互联网是由美国的ARPANET发展和演化而成的。ARPANET是全世界第一个分组交换网。1969年美国的国防高级研究计划署(DARPA)建立了一个只有4个节点的存储转发方式的分组交换广域网——ARPANET，该网是为了验证远程分组交换网的可行性而进行的一项试验工程。随后Bob Kahn和Vint Cerf合作，重新设计了**网络协议**，提出来作为互联网基础的TCP/IP协议，并于1983年在ARPANET上正式启用。TCP/IP协议简单、有效，成为了以后互联网得以迅速发展的关键因素之一。1983年ARPANET被分成两部分，一部分是专用于国防的Milnet网，剩下的部分以ARPANET为中心组成的新互联网称为Internet。为区别于一般的互联网，Internet的第一个英文字母用大写的I。从1969年ARPANET诞生到1983年Internet的形成是Internet发展的第一阶段，即研究试验阶段。

从1983年到1994年是互联网发展的第二阶段，其核心是NSF网的形成和发展，这是互联网在教育和科研领域广泛使用的实用阶段。1986年美国国家科学基金会(NSF)制订了一个使用超级计算机的计划，即在全美设置5个超级计算机中心，并建设一个高速主干网，把这些中心的计算机连接起来，形成NSF网。NSF网是一个3级分层的互联网，即NSF主干网、各个区域网和众多的校园网。NSF网的形成和发展，使它成为Internet的最主要的组成部分。与此同时，很多国家相继建立自己的主干网，并接入Internet，成为Internet的组成部分，如加拿大的CA * net，欧洲的EBONE和NORDUNET，英国的PIPEX和JANET，以及日本的WIDE等。

Internet最初的宗旨是用来支持教育和科研的活动，它不是营业性的商业组织，但是，随着Internet规模的扩大、应用服务的发展以及全球化需求的增长，出现Internet商业化的需求，并开始出现AlterNet网和PSI网(PSInet)等商用IP网络。为了解决商用IP网络接入Internet的问题，1991年出现了商用Internet交换(CIX)互连点，它由高速路由器和连接各CIX成员的链路组成，这些CIX的成员都是网络服务提供者，而不是网络用户。CIX使商业用户可以接入Internet，从此Internet的服务范围就不仅局限于教育、研究和政府部门了。1994年NSF宣布不再给NSF网运行维护经费支持，由MCI公司和Sprint公司运行维护，这样，不仅商业用户可以进入Internet，而且Internet的经营也商业化了。

Internet从研究试验的第一阶段到实用于科教的第二阶段，进而到商用的第三阶段的发展，反映了Internet技术和应用的成熟。20世纪90年代之后，随着网络技术的快速发展，Internet的规模迅速扩大，各种创新应用层出不穷。至2012年6月底，全世界Internet网民数量超过24亿，占世界人口比率为34.3%。更加重要的是，Internet提供了极为丰富的信息资源和应用服务。它为发展信息网络技术和网络应用提供了丰富的经验，对信息市场的开拓和信息社会的发展具有深远的影响，以Internet为基础的国家信息基础设施(NII)和全球信息基础设施(GII)正在形成。

1994年，我国互联网正式接入Internet，至2012年12月底我国Internet网民数达5.64亿，占人口比率为42.1%，成为全世界网民数最多的国家，Internet在我国社会生活各个方面起着十分重要作用。

随着IPv4地址空间已经耗尽，Internet正在向IPv6技术过渡，相关过渡技术的发展趋于成熟，发达国家纷纷提出了各自的过渡计划，我国也从2006年开始实施下一代互联网工程CNGI计划，启动IPv6试商用网络的建设。可以预见，在IPv4和IPv6这两种技术在Internet中将同时长期并存。

当前，Internet的全球化与商业化发展正面临技术、管理方面的众多挑战，在技术领域的挑战包括Internet的泛在问题、安全和信用管理问题、服务质量控制问题等；在管理领域的挑战包括管理规则和相应法律法规的适应性、竞争的引入与控制、知识产权管理等问题，这些挑战推动了在新一代互联网体系结构和治理模式方面的不断探索。在互联网体系结构的探索方面，出现了包括内容中心网络(content centric network)在内的一些新的体系结构概念，并推动了软件定义网络(software defined network)等新网络技术的发展。对Internet泛在问题的探索推动了物联网技术、社交网络技术及其应用的发展，对人

类社会正产生重大影响。在联合国的组织下,以峰会的形式开始了世界各国对 Internet 治理模式的讨论,以推动其向更为公平公开的方向发展,使之有益于打破垄断和消除数字鸿沟。

参考文献

1. Leiner B M. Cerf V G. A brief history of the Internet. ISOC, 1997

2. Comer D E. Internetworking with TCP/IP. Englewood Cliffs: NJ: Prentice Hall Inc., 1993

3. Russel J, Cohn R. ICANN. Book on Demand Ltd., 2013

(胡道元　龚俭)

hulianwang dizhi

互联网地址(Internet address)　用来标识 Internet 上每台计算机的二进制数,又称 **IP 地址**。

在 IP 协议中(参见 **TCP/IP 协议集**),规定分配给每台主机一个二进制数作为该主机的 IP 地址。在 Internet 上发送的每个数据报头都包含了 IP 源地址和 IP 目标地址。

根据 IP 协议的版本号不同,IP 地址的长度不同。IPv4 地址为 32 位二进制数,为了方便可用一种点分十进制表示法来表示,将 32 位二进制数中每 8 位为一组,用十进制表示,利用小圆点分割各部分。每段中最小值为 0,即一组内的所有位都为 0,最大值为 255,即组内所有位数都为 1。IPv6 地址是 128 位二进制数,为了方便以每 16 位为一组,用一种冒号划分的十六进制表示法来表示。为了便于书写,其中连续的 0 可以用::表示,如 2001:db8:0:0:0:0:0:1 可以表示为 2001:db8::1。IP 地址分为单播地址、组播地址和其他地址类型。

单播描述点对点的通信过程,其源地址和目标地址均为单播地址。每个单播 IP 地址由两部分组成,即网络标识和主机标识,网络标识确定了该台主机所在的物理网络,主机标识确定了在某一物理网络上的一台主机。在 IPv4 协议中,早期把单播地址空间划分为 A、B、C 三种基本类,每类有不同长度的网络标识和主机标识。随着无分类地址的引入,目前 IPv4 协议和 IPv6 协议的网络标识和主机标识均由前缀长度决定,如图 1 所示。

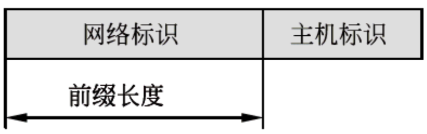


图 1　网络标识和主机标识

组播描述点对多点的通信过程,其源地址为单播地址,目标地址为特殊定义的地址段,称为组播地址。任何需要接收发到某个组播地址的数据报需要动态加入该组播地址。IPv4 的组播地址范围为 224.0.0.0 到 239.255.255.255,IPv6 的组播地址范围为 ff00::/8。

将一台计算机的 IP 地址绑定到物理地址的过程称地址解析。一台主机或路由器在需要向同一物理网络上的另一台计算机发送数据时,先要进行地址解析。IPv4 的地址解析协议为 ARP,IPv6 协议的地址解析协议称为邻居发现协议(ND)。

不同 IPv4 地址之间或 IPv4 地址到 IPv6 地址的映射称为地址翻译,翻译的过程可以是有状态的翻译(参见**网络地址翻译**),也可以是无状态的翻译,即要求 IPv4 地址内嵌在 IPv6 地址中。

参考文献

1. Comer D E. Internetworking with TCP/IP, Volume Ⅰ. 3rd ed. Englewood Cliffs, NJ: Prentice Hall, 1995

2. Fuller V, Li T. RFC4632: Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan. 2006

(李星　包丛笑　胡道元)

hulianwang fuwu gongyingshang

互联网服务供应商(Internet service provider, ISP)　指向个人或企业用户提供接入互联网、提供互联网基础信息服务或增值服务的网络运营商。

ISP 使用各种通信技术向个人或企业用户提供接入互联网的服务,仅提供这种服务的 ISP 有时又被称为互联网接入服务供应商 IAP(Internet access provider)。ISP 为个人用户提供的接入手段包括利用固定电话线路或移动电话拨号上网,利用有线电缆(以太网、电视同轴电缆)、光纤或无线局域网 WLAN、高速无线通信系统等。ISP 为企事业单位接入互联网时,则根据需要提供时分复用的专用信道、以太网或者直通光缆等多种速率和方式的接入服务。

尽管没有权威定义,行业里普遍认为根据网络互通互联能力、服务网络规模、覆盖地域以及是否可以不经付费购买网络互联服务就可以通达全球网络等差别,互联网接入服务供应商还分为第一级(Tier 1)和第二级(Tier 2)、第三级(Tier 3)等不同级别。

不同级别的 ISP 之间构成对等互联关系或接入关系。

互联网接入服务供应商还可以提供主机托管服务。通过签订协议为个人或中小型企业提供利用集中管理的计算机服务器集群实现电子邮件、域名解析、文件共享、视频流媒体以及基于 Web 的信息内容等网站接入互联网的服务。（马严）

hulianwang gongcheng renwuzu

互联网工程任务组（Internet Engineering Task Force，IETF） ISOC（互联网协会）下属的网络标准化组织，也是网络运营商、硬件和软件厂商和研究者在一起研讨并完善互联网协议、标准和产品的论坛。IETF 涉及的标准化工作涵盖从 IP（互联网协议）到通用应用之间的所有协议层。IETF 的讨论内容不涉及传输硬件，这部分标准化工作由 IEEE 和 ITU 完成；IETF 也不涉及特定应用层协议，例如 HTML（超文本标记语言）和 XML（扩展标记语言）的标准化工作由 W3C 完成。

成立于 1986 年的 IETF 是唯一不采用会员制的国际著名标准化组织，没有会员费，不需要签署任何的参加协议。任何参与者默认接受 IETF 的规则，包括知识产权方面的规则。任何参与者只代表其个人，不代表他服务的机构或公司。由于没有会员制，IETF 任何发布标准都没有投票环节，IETF 的标准制定和发布主要依赖于专业人员的共识和实际系统的验证。IETF 的活动是开放的，全球所有对互联网标准制定感兴趣的专业人员均可以参加工作组的邮件列表的讨论并有选择地参加 IETF 举办的一年三次会议，参与互联网标准的研究和制定工作。

IETF 的成果是免费出版的 RFC（征求意见）文档。RFC 这个名称意味着互联网是一个不断变化的技术系统，任何一个 RFC 都可能会被后出现的新 RFC 替代。只有“标准类”或“最佳当前实践”类的 RFC 才属于 IETF 正式批准的标准。“信息”“实验”和“历史”类的 RFC 都不属于 IETF 标准。标准的 RFC 有四种形式：建议标准（PS），第一个正式阶段的标准；草案标准（DS），第二个正式阶段的标准，说明已经完成了互操作性实现的演示；标准（STD），最后阶段，表明标准已经被广泛部署；最佳当前实践（BCP），不同于以上三个阶段的单一阶段标准，更侧重于操作规范。

IETF 的标准研制工作分成 8 个领域：应用领域（APP），主要侧重于普遍使用的应用协议，互联网基础设施协议以及可在其他协议中重用的构件协议；通用领域（GEN），研究支持、更新和维护 IETF 标准研发过程等；互联网领域（INT），研究 IP 层、不同 IP 版本的共存、域名服务、移动性等协议；运营和管理领域（OPS），研究网络管理、互联网运营等方面的技术标准；实时应用与基础设施领域（RAI），研究延迟敏感的人际通信协议和体系结构；路由领域（RTG），维护已有路由协议的可缩放性和稳定性，开发、扩展和修复协议，确保互联网路由系统的持续运行；安全领域（SEC），研究提供完整性、身份认证、保密性等服务的安全协议和密钥管理机制；传送领域（TSV），研究互联网中与端到端数据传送相关的技术标准。

每个领域有一个或多个领域主管，这些主管构成了互联网工程指导组（IESG）。IESG 负责 IETF 活动组织和互联网标准管理，包括增加、合并、关闭或重定义 IETF 的研究领域，批准互联网标准。每个领域通常有多个工作组，它是互联网标准研究和制定的最基本组织。每个工作组通过工作组章程确定工作的目标和阶段性工作计划，一旦工作组完成了目标，发布了相关的文档，就可以终止该工作组。IETF 工作的监管由互联网体系结构委员会（IAB）执行。

参考文献

http://www.ietf.org/　（沈苏彬）

hulianwang kongzhi baowen xieyi

互联网控制报文协议（Internet Control Message Protocol，ICMP） 运行 TCP/IP 协议族的 IP 网络中的一个基本协议，借助于 IP 协议在终端系统（如 IP 主机）和网络设备（如路由器或网关）之间传递相关网络链路状态的消息。需要说明的是，对应不同的 IP 版本（IPv4 和 IPv6），ICMP 也有不同的版本（ICMPv4 和 ICMPv6），除 IP 地址空间有所变化外（32 位和 128 位），ICMP 报文携带的参数也有所不同。

ICMP 报文分为两类：差错报告和信息报文。

针对 IP 协议提供的不可靠传输服务，ICMP 差错报告报文用于反映 IP 报文在 IP 网络传递过程中遇到的问题，也即当路由器或终端系统中的 IP 实体（执行 IP 协议的程序）在转发或投递 IP 报文时发现故障，它在丢弃该 IP 报文的同时，也会向 IP 报文的源发端系统反馈一个 ICMP 差错报文，通知其故障及原因，例如可能是目的 IP 地址错误，或者 IP 报文体积超出承载网络的数据单元体积，且不容许分段，

而不得不被丢弃。ICMP 报文的传递依赖于 IP 协议,作为 IP 报文的负载在 IP 网络中传递。如果承载 ICMP 报文的 IP 报文出现无法转发或者投递时,会被直接丢弃,并不再产生任何差错报告。

ICMP 信息报文则用于辅助用户获取网络的相关信息,如收到回应请求(Echo)ICMP 报文的终端系统/路由器必须用回应应答(Echo Reply)ICMP 报文予以响应,以体现响应者的工作状态;收到时戳请求(TimeStamp)ICMP 报文的终端系统/路由器必须用时戳应答(TimeStamp Reply)ICMP 报文予以响应,通过报文中记录的原始时戳(请求报文的发出时间)、接收时戳(收到报文请求的时间)和传输时戳(发出回应报文的时间)来反映网络和中间设备的拥塞情况等。

ICMP 的应用很广,除了辅助用户了解 IP 报文的投递结果以外,还常被用户用来开发网络状态检测工具,如 Unix 操作系统中探测节点是否上线的 Ping 命令和用于追踪报文传递路径的 Traceroute 命令等。

参考文献

1. Postel J. Internet control message protocol. STD 5, RFC 792, September 1981

2. Conta A, Deering S, Gupta M. Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification, RFC 4443. March 2006

(吴国新)

hulianwang neirong gongyingshang

互联网内容供应商(Internet content provider, ICP)　向互联网用户综合提供包括搜索引擎、即时通信、虚拟社区、电子邮箱、新闻娱乐、网络游戏等互联网信息和增值业务的运营商。又称互联网网络内容服务商。(马严)

hulianwang tixi jiegou

互联网体系结构(Internet architecture)　Internet 采用的协议及服务的概念性模型和结构。Internet 采用 TCP/IP 协议(参见 **TCP/IP 协议集**),故又称为 TCP/IP 互联网。Internet 体系结构也是一种分层模型。它由 5 个层次构成,即应用层、传输层、IP 层(网络层)、网络接口层(数据链路层)和物理层(硬件)。图 1 给出了这些概念性层次结构以及这些层次之间传送数据的形式。

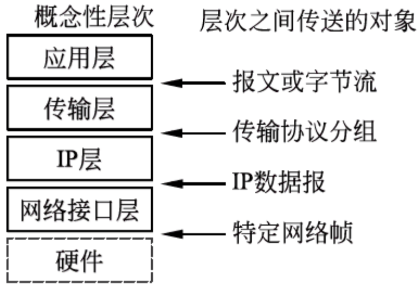


图 1　概念性层次结构

(1) 应用层　应用层提供用户可使用的服务,用户调用应用程序来访问 TCP/IP 互联网络提供的这些服务。应用程序负责发送和接收数据。每个应用程序可选择所需的传送服务类型,既可以传送独立的报文序列,也可以传送连续的字节流。应用程序的任务是将数据按要求的格式传送给传输层。

(2) 传输层　传输层的基本任务是提供发送端和接收端的应用层之间的通信,即端到端的通信。传输层管理信息流,提供可靠的传输服务,以确保数据无差错地、按序地到达。传输层软件将要传送的数据流划分成组,并连同目的地址传送至下一层。

(3) IP 层　IP 层处理机器之间的通信,它接收来自传输层的请求,将带有目的地址的分组发送出去。IP 层将分组封装到数据报(参见**网际协议**)中,填入数据报头,使用路由算法以决定是直接将数据报传送至目的主机还是传给路由器,然后把数据报送至相应的网络接口来传送。IP 层还处理接收到的数据报,检验其正确性,并决定是由本地接收还是通过路由器传送至相应的目的站。

(4) 网络接口层　也称数据链路层,该层负责接收 IP 数据报并发送至选定的网络。网络接口包括一个设备驱动器,也可能是一个复杂的具有数据链路协议的子系统。

(5) 物理层(硬件)　协调在物理介质上传输位流所需的各功能。

Internet 体系结构的概念性层次包含两个重要的分界线:一个是协议地址分界线,以区分高层和低层的寻址;另一个是操作系统分界线,以区分系统与应用程序。如图 2 所示。

高层寻址使用 IP 地址,低层寻址使用物理地址。应用程序 IP 层之上的所有协议软件只使用 IP 地址,而网络接口层处理物理地址。

通常将软件分成操作系统和非操作系统软件两部分。当协议软件集成到操作系统中后,在协议软

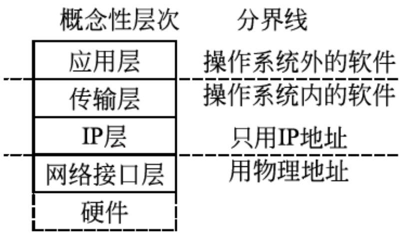


图 2　概念性层次的分界

件的低层之间进行数据传送的开销比应用层和传输层之间进行数据传送的开销要小得多。

TCP/IP 的分层工作原理如图 3 所示,表示两台主机上的应用程序之间传输报文的路径。主机 B 上的第 n 层接收的正是主机 A 上的第 n 层发送出来的对象。在物理网络上传送的帧是相同的帧,以上各层收发的分别为相同的数据报、相同的分组和相同的报文。

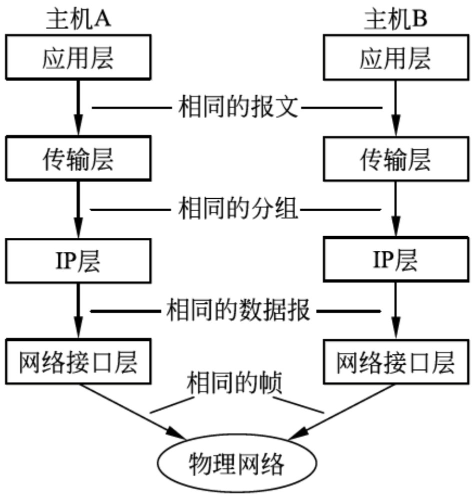


图 3　TCP/IP 分层工作原理

Internet 软件是围绕着三个层次的概念化网络服务设计的。最基本的互联网服务由一个分组传送系统组成,该服务被定义为不可靠的、无连接的、尽最大努力传送的分组传送系统。所谓不可靠,指的是不能保证正确传送,分组可能丢失、重复、延迟或不按序传送,而且服务不检测这些情况,也不通知发送方和接收方。所谓无连接,指的是每个分组都是独立处理的,可能经过不同的路径,有的可能丢失,有的可能到达。所谓尽最大努力传送,指的是互联网软件尽最大努力来传送每个分组,只有当资源用尽或底层网络出现故障时,才会出现不可靠服务。

这种不可靠的、无连接传送的协议称为网际协议,简称 IP 协议。

参考文献

1. 胡道元. 计算机网络. 2 版. 北京:清华大学出版社, 2009

2. Comer D E. Internetworking with TCP/IP, Volume Ⅰ. 4th ed. Englewood Cliffs, NJ: Prentice Hall, 2000

(胡道元)

hulianwang yingyong gongyingshang

互联网应用供应商(application service provider, ASP)　通过互联网为各种各样的商务和事业客户提供其所需的应用服务的运营商。互联网应用供应商是通过托管或者租用的形式实现对客户的服务,而不是使用传统的购买或者定制开发的形式提供,从而使客户的需求得以实现快速满足并大幅度降低开发和运行维护成本。例如把文字处理等软件放在网络上,供所有的在线用户通过网络来使用这些软件就是一种 ASP 业务。由此用户就可以非常清晰地感觉到 ASP 就是通过互联网络为客户提供所需的服务。

与传统的专业外包服务相似,互联网应用供应商实际是这些传统专业外包服务在互联网上的延伸。所有应用服务产品被放置在互联网应用供应商的数据中心供其签约客户随时调用,同时由互联网应用供应商动态地进行管理、维护并更新这些服务产品。通过将各种软件、硬件、网络和专业技术的合理搭配,向客户提供更优质完善的服务。与传统的公司内部运作的应用软件和服务相比较,这种服务更安全、可靠、经济,具有更灵活的可扩展性。这种服务方式可以使中小企业特别是小企业能够得到过去只能由中大型企业才具有的专业能力。　(马严)

huan

环(ring)　一种具有两个二元运算的代数结构。

设 R 为非空集合且在 R 上已定义了两个二元运算加法“+”和乘法“*”。如果 R 关于“+”构成交换群,“*”对“+”满足分配律,即对任意 $a,b,c\in R$ 有 $a*(b+c)=(a*b)+(a*c)$ 和 $(a+b)*c=(a*c)+(b*c)$,则称 R 为一个**非结合环**,记为 $\langle R.eps,+,*\rangle$。

在非结合环 R 中,有唯一的零元“0”且每个 $a\in R$ 都有唯一的负元“$-a$”,即对任意 $a\in R$ 有 $a+0=a$ 和 $a+(-a)=0$。常把 $a*b$ 和 $a+(-b)$ $(a,b\in R)$ 分别写成 ab 和 $a-b$,并约定“*”优先于

“+”结合。因此,对任意 $a,b,c\in R$ 和整数 n,恒有

$$a0=0=0a,\quad a(-b)=-ab=(-a)b$$
$$(-a)(-b)=ab,\quad a(b\pm c)=ab\pm ac$$
$$(a\pm b)c=ac\pm bc,\quad (na)b=nab=a(nb)$$

还可由归纳法证明

$$\left(\sum_{i=1}^{n}a_i\right)\left(\sum_{j=1}^{m}b_j\right)=\sum_{i=1}^{n}\sum_{j=1}^{m}a_ib_j$$
$$a_1,a_2,\cdots,a_n,b_1,b_2,\cdots,b_m\in R\quad 且\ n,m\geqslant 1$$

设 R 为非结合环。若对任意 $a,b,c\in R$ 有 $a^2=0=(ab)c=(bc)a=(ca)b$,则称 R 为一个李环。若对任意 $a,b\in R$ 有 $((aa)b)a=(aa)(ba)$ 和 $ab=ba$,则称 R 为一个若尔当环。李环和若尔当环是非结合环中内容最丰富的分支。

如果非结合环 R 满足结合律,即对任意 $a,b,c\in R$ 有 $a(bc)=(ab)c$,则称 R 为一个环。如果环 R 满足交换律,即对任意 $a,b\in R$ 有 $ab=ba$,则称 R 为**交换环**。

设 R 为环。若 $a,b\in R\{0\}$ 使 $ab=0$,则称 a 为 R 的左零因子,b 为 R 的右零因子,它们统称为 R 的零因子。若 $e\in R$ 使得对任意 $a\in R$ 有 $ae=a=ea$,则称 e 为 R 的一个幺元或单位元。环 R 的幺元若存在,必是唯一的。如果环 R 具有幺元和非零元,R 中任意非零元 a 有逆元 a^{-1},即 $aa^{-1}=e=a^{-1}a$,则称 R 为一个体。交换体称为域。

全体整数、有理数和实数关于普通的加法和乘法都构成交换环,并分别称为整数环、有理数环和实数环。这些环都无零因子。

设 m 为一个正整数。模 m 的全体剩余类 $\{\overline{0},\overline{1},\cdots,\overline{m-1}\}$ 关于模 m 的加法 $+_m$ 和模 m 的乘法 $*_m$ 构成一个交换环,称为模 m 的剩余类环,记为 $\mathbf{Z}/(m)$。当 m 为合数时,$\mathbf{Z}/(m)$ 有零因子。

设环 R 具有幺元,$x\notin R$ 是未定元。若 $n\geqslant 0$ 且 $a_0,a_1,\cdots,a_n\in R$,则称 $a_0+a_1x+\cdots+a_nx^n$ 为一个 R 上 x 的多项式。全体 R 上 x 的多项式关于多项式的加法和乘法构成一个环,称为 R 上 x 的**多项式环**,记为 $R[x]$。

设 D 为环 R 的一个非空子集。若对任意 $a,b\in D$ 和 $r\in R$ 恒有 $a-b,ra,ar\in D$,则称 D 为 R 的一个理想。对任意 $a,b\in R$ 必有

$$(a+D)\oplus(b+D)\triangleq\{(a+d)+(b+d')\mid d,d'\in D\}=(a+b)+D$$
$$(a+D)\circledast(b+D)\triangleq\{(a+d)(b+d')\mid d,d'\in D\}=ab+D$$

因此 $\{a+D\mid a\in R\}$ 关于上面定义的二元运算 $\oplus$ 和 $\circledast$ 构成一个环,称为 R 关于 D 的**商环**,记为 R/D。

设 R 和 R' 为两个环且 $f:R\to R'$。若对任意 $a,b\in R$ 有 $f(a+b)=f(a)+f(b)$ 和 $f(ab)=f(a)f(b)$,则称 f 为一个**环同态**,并称 $\ker(f)=\{a\in R\mid f(a)$ 为 R' 的零元$\}$ 为 f 的核。若环同态 f 是双射,则称 f 为一个**环同构**,并称 R 与 R' 同构,记为 $R\cong R'$。

设 R 和 R' 为两个环。若 $f:R\to R'$ 为环同构,则 $\ker(f)$ 为 R 的理想,而且当 $f(R)=R'$ 时有 $R/\ker(f)\cong R'$(环同态定理)。

参考文献

1. Hungerford T W. 代数学. 冯克勤,译. 长沙:湖南教育出版社,1985

2. 熊全淹. 近世代数. 上海:上海科技出版社,1978

(王兵山　王水汀)

huigui fenxi

回归分析(regression analysis)　研究一个或多个随机变量与若干非随机变量之间关系的统计方法。考虑一个可观测的随机变量 y 与 p 个可控制的变量 $x_1,x_2,\cdots,x_p$ 有某种函数关系,假定函数的形式未知或者函数形式已知,但其中含有某些未知的参数 $\beta_1,\beta_2,\cdots,\beta_p$,于是有 $y=f(x_1,\cdots,x_p,\beta_1,\cdots,\beta_p)+\varepsilon$,其中 ε 为随机的观测误差,称 $f(x_1,\cdots,x_p,\beta_1,\cdots,\beta_p)$ 为 y 关于 $x_1,\cdots,x_p$ 的回归。通常假定 f 为 $x_1,\cdots,x_p$ 的线性函数,这是因为在某种情形下,比如 $y=a\sin t+b\cos u$,只要令 $x_1=\sin t,x_2=\cos u$,仍可化为线性问题来研究。另一方面对于一般光谱的函数 f,总可以用多项式来逼近,作适当的变换也可化为线性问题。所以可设

$$y=\beta_0+\beta_1x_1+\cdots+\beta_{p-1}x_{p-1}+\varepsilon \tag{1}$$

写成矩阵的形式

$$\boldsymbol{y}=\boldsymbol{X\beta}+\boldsymbol{\varepsilon} \tag{2}$$

其中 $\boldsymbol{y}=(y_1,y_2,\cdots,y_n)^{\mathrm{T}}$,$\boldsymbol{\beta}=(\beta_0,\beta_1,\cdots,\beta_{p-1})^{\mathrm{T}}$,$\boldsymbol{\varepsilon}=(\varepsilon_1,\varepsilon_2,\cdots,\varepsilon_n)^{\mathrm{T}}$

$$\boldsymbol{X}=\begin{bmatrix}1 & x_{11} & \cdots & x_{1\,p-1}\\ \vdots & \vdots & \vdots & \vdots\\ 1 & x_{n1} & \cdots & x_{n\,p-1}\end{bmatrix}$$

为 $n\times p$ 阶的已知矩阵,$\mathrm{E}\,\boldsymbol{\varepsilon}=0$,$\mathrm{var}(\boldsymbol{\varepsilon})=\sigma^2\boldsymbol{I}_n$。这里所谓线性是对参数 $\boldsymbol{\beta}$ 而言的。

回归分析的主要问题是:①如何估计 $\beta_0,\cdots,\beta_{p-1}$ 以及 σ^2;②对有关 $\boldsymbol{\beta}$ 的某种假设和模型(2)的假设进行检验;③对 $\boldsymbol{y}$ 进行预测或控制。

参数的最小二乘估计　考虑误差平方和

$$Q = \boldsymbol{\varepsilon}^{\mathrm{T}}\boldsymbol{\varepsilon} = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

其中 $\hat{y}_j = \sum_{i=0}^{\beta-1}\beta_i x_{ji}, j = 1,2,\cdots,n$ 称为估计值。使得 Q 达到最小的估计称为最小二乘估计，它们是

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y}, \quad \hat{\sigma}^2 = \frac{1}{n}Q$$

它们都是极大似然估计，而 $\hat{\sigma}_e^2 = \frac{1}{n-p}Q$ 为 σ^2 的无偏估计。

对 y 的预测与控制　给定置信水平 $1-\alpha$，令

$$\left.\begin{aligned}\hat{\boldsymbol{y}}_1 &= \hat{\boldsymbol{y}} - \hat{\sigma}_e t_{n-p}(\alpha)\Big(1 + \sum_{i=1}^{p}\sum_{j=1}^{p}c_{ij}\boldsymbol{x}_i\boldsymbol{x}_j\Big)^{\frac{1}{2}} \\ \hat{\boldsymbol{y}}_2 &= \hat{\boldsymbol{y}} + \hat{\sigma}_e t_{n-p}(\alpha)\Big(1 + \sum_{i=1}^{p}\sum_{j=1}^{p}c_{ij}\boldsymbol{x}_i\boldsymbol{x}_j\Big)^{\frac{1}{2}}\end{aligned}\right\} \tag{3}$$

则有 $p(\hat{\boldsymbol{y}}_1 \leqslant \boldsymbol{y} \leqslant \hat{\boldsymbol{y}}_2) = 1-\alpha$，式中 $\hat{\boldsymbol{y}} = (\hat{y}_1, \hat{y}_2, \cdots, \hat{y}_n)^{\mathrm{T}}$，$\hat{y}_j = \sum_{i=1}^{p}\hat{\beta}_i x_{ji}$，$\boldsymbol{x}_i$ 为 $\boldsymbol{X}$ 中第 i 个列向量，c_{ij} 为 $(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}$ 中元素，$t_{n-p}(\alpha)$ 为 t 分布的临界值。由此，如要控制 $\hat{\boldsymbol{y}}_1, \hat{\boldsymbol{y}}_2$ 的值，便可确定相应的 $\boldsymbol{x}_i$ 的值。

线性模型的假设检验　检验所提的线性模拟是否合适。为此令 $H_0: \beta_1 = \beta_2 = \cdots = \beta_p = 0$。如果 H_0 不被否定则说明模型不合适或拟合的效果甚差。由于 H_0 成立时，统计量 $(n-p)U/(pQ_e) \sim F(p, n-p)$ 分布（这里 $U = \boldsymbol{y}^{\mathrm{T}}\boldsymbol{X}(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y}, Q_e = \boldsymbol{y}^{\mathrm{T}}(\boldsymbol{I} - \boldsymbol{X}(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{T}})\boldsymbol{y}$），于是得到拒绝域为 $(n-p)U/pQ_e > F_{p,n-p}(\alpha)$，这里 $F_{p,n-p}(\alpha)$ 是 $F_{p,n-p}$ 分布的临界值。

为检验某个自变量（如第 i 个）对 y 的影响甚微，可提出 $H_0: \beta_i = 0$，在 H_0 成立下，$F \triangleq (n-p)\beta_i^2/(Q_e c_{ii}) \sim F_{1,n-p}(\alpha)$。由此可确定拒绝域为 $F > F_{1,n-p}(\alpha)$。

举一个例子，有人从钢线的碳含量 x（%）与 20℃时的电阻值 y（μΩ）的测量数据中，得到回归方程

$$\hat{y} = 13.9584 + 12.5503x$$

于是便可预测，当碳含量为 50% 时，钢线的阻值以 95% 的置信度处于 19.66 μΩ 到 20.81 μΩ 之间。

参考文献

陈希孺. 数理统计引论. 北京：科学出版社，1981

（金治明）

huisu fa

回溯法（backtracking approach）　一种一般的算法设计方法。回溯法可用于求特解、解集和最佳解，其基本思想是：按照带约束函数的深度优先结点生成法来形成状态空间树。即：按照深度优先的方式从活结点表中选取扩展结点进行扩展，并且及时使用约束函数将不可能产生所需解的活结点变成死结点。所谓扩展结点就是一个正在产生儿子的结点。活结点就是一个自身已产生但其儿子还没有全部生成的结点。死结点就是一个所有儿子已经产生的结点。在状态空间树中，每个结点表示一个问题状态。从根到任何结点的全部路径形成问题的状态空间。解状态是这样的结点，使得从根到该结点的路径定义了解空间中的一个多元组。所有解状态的集合称为解状态集。回答状态集是解状态集的一个子集，从根到其元素的路径定义了解空间中满足全部约束条件的一个解。因此，问题求解过程可看成首先生成问题状态，然后确定哪些问题状态是解状态，最后确定哪些解状态是回答状态。

在使用回溯法求解问题时，首先要将问题的解用向量的形式来表示，即：定义解向量的形式 $(x_1, x_2, \cdots, x_n)$。然后确定显约束，即：对 $1 \leqslant i \leqslant n$，给定 x_i 的取值范围，如：$x_i \in S_i$，显约束一般依赖于问题的输入 I。它确定关于 I 的解空间。对 $1 \leqslant i \leqslant n$，若 $|S_i| = m_i$，则解空间中解向量的数量为：$m = m_1 m_2 \cdots m_n$。最后还要确定隐约束，即：给定 x_i 之间的关系，如满足约束函数 $B(x_1, x_2, \cdots, x_n)$。这时，若部分向量 $(x_1, x_2, \cdots, x_i)$，$1 \leqslant i \leqslant n$ 无法导出解向量，则不考虑 $x_{i+1}, x_{i+2}, \cdots, x_n$ 的任何选择。进一步，在确定了部分解向量 $(x_1, x_2, \cdots, x_i)$ 之后，若 $x_{i+1} \in S_{i+1}(x_1, x_2, \cdots, x_i)$ 且 $B_{i+1}(x_1, x_2, \cdots, x_{i+1})$ 为真，则继续扩展，否则，不考虑这个 x_{i+1}。这里，$x_{i+1} \in S_{i+1}(x_1, x_2, \cdots, x_i)$ 和 $B_{i+1}(x_1, x_2, \cdots, x_{i+1})$ 分别是由 $x_{i+1} \in S_{i+1}$ 和 $B(x_1, x_2, \cdots, x_n)$ 派生出来的约束条件。

下面是使用回溯法求解问题时的一个一般的非递归算法框架。其中 n 是解向量 X 的维数，生成值 $X(k)$ 的顺序将影响算法的效率。

```
procedure BACKTRACK(n)
 begin
  k←1;//k 为当前考察的分量的下标
  while k>0 且 k≤n do
  if 对于还没有试探过的值 X(k)，有 X(k) ∈
    S_k(X(1),X(2),…,X(k-1))且 B_k(X(1),
    X(2),…,X(k))的值为真
  then
```

```
  begin
    if (X(1),X(2),…,X(k))是一个解 then
      write (X(1),X(2),…,X(k));
    if k<n then k←k+1//深度优先
  end
 else k←k-1//回溯
end
```

下面是使用回溯法求解问题时的一个一般的递归算法框架。其中参数 k 为当前考察的分量的下标,全局量 n 是解向量 X 的维数,用于控制递归深度。执行 RBACKTRACK(1)可求全部解。

```
procedure RBACKTRACK(k)
 if k>n then return else
 for 每个使 B_k(X(1),X(2),…,X(k))的值为真的
    X(k)∈S_k(X(1),X(2),…,X(k-1)) do
    begin
      if (X(1),X(2),…,X(k))是一个解 then
      write (X(1),X(2),…,X(k));
      RBACKTRACK(k+1)//深度优先
    end//通过递归调用的返回来实现回溯
```

针对具体问题应用回溯法时,若约束函数比较复杂,则可将对约束函数的检查单独写成子程序。若显约束集 S_i 中元素数目较少,则可将框架中的循环变成分情况讨论。

使用回溯法求解问题时,若解向量的形式为 $(x_1,x_2,\cdots,x_n)$,为了加速执行,一般按照重排原理来确定各分量考察的顺序,即优先考察 $|S_i|$ 最小的分量 x_i。一般来说,回溯法的时间复杂性与穷举法的时间复杂性量级相同,但前者的比例因子小。具体来说,回溯法的时间复杂性可按状态空间树中结点的总数乘以处理每个结点所需时间来估算。假设第 $h-i$ 层(根结点在第 h 层)的每个约束结点(即满足约束条件的结点)有 m_{i+1} 个儿子,那么第 h 层有 1 个结点,第 $h-1$ 层有 m_1 个结点,第 $h-2$ 层有 m_1m_2 个结点,第 $h-3$ 层有 $m_1m_2m_3$ 个结点,…,第 0 层有 $m_1m_2m_3\cdots m_h$ 个结点,因此,状态空间树中约束结点的总数为

$$m = 1 + m_1 + m_1m_2 + m_1m_2m_3 + \cdots + m_1m_2m_3\cdots m_h$$

下面的过程 ESTIMATE 估算了状态空间树中约束结点的总数,其中,SIZE(S)返回集合 S 中元素的数目,CHOOSE(S)随机地返回集合 S 中的一个元素。为了获得较为准确的估算,可多次运行该过程,然后以输出结果的均值作为估算的结点总数。

```
procedure ESTIMATE
 begin
  m←1; r←1; k←1;
  while k≤n 且 S(k) =
      {X(k)∈S_k(X(1),X(2),…,X(k-1)) |
      B_k(X(1),X(2),…,X(k))}非空 do
    begin
      r←r * SIZE(S(k)); m←m+r; X(k)←
      CHOOSE(S(k)); k←k+1
    end;
  write(m)
 end
```

作为一种一般的算法设计方法,回溯法可用于求解各种各样的问题(如 N 后问题、子集和问题、图的可着色性问题、哈密顿回路问题等)。

参考文献

1. Aho A V, Hopcroft J E, Ullman J D. The design and analysis of computer algorithms. Boston, MA: Addison-Wesley, 1974

2. Cormen T H, Leiserson C E, Rivest R L, et al. Introduction to algorithms. 3rd ed. Cambridge, MA: The MIT Press, 2009

3. Kleinberg J, Tardos E. Algorithm design. Boston, MA: Addison-Wesley, 2005

(殷建平)

huibian chengxu

汇编程序(assembler) 把**汇编语言**书写的源程序翻译成等价的**机器语言**程序的处理系统。

汇编程序以汇编语言书写的源程序作为输入,以机器语言表示的目标程序作为输出,其主要工作过程是:①输入汇编语言源程序;②检查语法的正确性,如果正确,则将源程序翻译成等价的二进制或浮动二进制机器语言程序,并根据用户的需要输出源程序与目标程序的对照清单;如果语法有错,则输出错误信息,指明错误的部位、类型和编号;③对翻译出的目标程序进行必要的善后处理,如直接启动执行,以可执行程序段形式保存在外存中,与外存中其他程序段或程序库中的程序连接装配成完整的目标程序。

在广泛使用汇编程序的过程中,人们不断地吸收其他处理系统的优点,相继研制出*模块汇编程序*、*宏汇编程序*、*高级汇编程序*、*条件汇编程序*等各具特色的汇编程序。模块汇编程序吸收了模块程序设计

思想,提供模块单独汇编功能,支持模块的并行设计、编程、调试和连接装配等能力。宏汇编程序的特点是在汇编程序的基础上增加宏加工程序功能,允许用户在源程序中方便地定义和使用宏指令。高级汇编程序的基本思想是用汇编语言编写源程序中的数据加工部分,而用高级语言的控制语句编写控制部分。这样既保持了汇编语言的优点,又吸收了高级语言易于书写和阅读的长处。条件汇编程序允许用户在源程序中设置条件汇编指示,并通过预置不同参数值的方法灵活地剪裁(选择或跳过)、汇编不同的程序段。

鉴于汇编语言的指令与机器语言的指令大体上保持一一对应的关系,所以汇编程序通常采用两遍扫描源程序的实现算法。第一遍查明源程序中符号的定义和使用情况,并将有关信息收集到符号表中;第二遍利用符号表中的信息,将源程序中的符号化指令逐条翻译成相应的机器指令。如果汇编语言中规定“所有符号一定要先定义,后使用”,则可以容易地将两遍算法合并成一遍算法实现相应的汇编程序。汇编程序的具体翻译工作可归纳为如下几项:①用机器操作码代替符号化的操作符;②用数值地址代替符号名字;③将常数翻译为机器的内部表示;④分配指令和数据所需的存储单元。除上述翻译工作外,汇编程序还应考虑如下工作:①处理汇编伪指令,收集程序中提供的汇编指示信息,并执行相应的功能;②向用户提供汇编过程中的有关信息以及源程序与目标程序的对照清单;③根据用户要求和汇编后程序段的特点,执行不同的善后处理工作,如直接执行、记入外存保留、连接装配等;④如果汇编语言支持模块汇编、宏汇编、高级汇编或条件汇编等功能时,汇编程序则应提供相应的处理任务。

(曹东启)

huibian yuyan

汇编语言(assembly language) 将**机器语言**中地址部分符号化并可进一步包括宏构造的语言。

汇编语言由汇编指令和汇编伪指令两部分组成。汇编指令是机器指令的地址部分符号化表示,其操作码采用易记的操作符表示,而地址码则采用标号、变量名字、常数等直观的表示形式。汇编指令基本上与机器指令保持一一对应的关系。在汇编过程中,它将被翻译成对应的机器指令;运行时,它将执行相应机器指令所规定的功能。汇编伪指令又称作汇编指示,其作用是指示汇编程序如何进行汇编,用于向汇编程序提供用户自定义的符号、数据类型、数据空间长度、目标程序格式、数据或指令的存放位置等提示信息。采用汇编语言编写程序虽不如高级语言简单、直观,而且必须对机器内部结构和特性有较多的了解;但它占用内存少,运行效率高,且能直接控制各种设备资源。因此,汇编语言经常用于编写大型软件系统的核心部分程序,或者用于编写运行时间长或实时性要求高的程序部分。

在不断吸收先进编程思想和相关软件优点的基础上,发展、演变出各具特色的汇编语言。模块汇编语言是在模块程序设计思想指导下,以模块作为编程基本单位而设计的汇编语言,它支持单个模块独立汇编和多个模块联合汇编的功能。宏汇编语言是在汇编语言的基础上,增加宏定义和宏调用功能而形成的汇编语言,它将为用户提供自定义指令的功能。高级汇编语言是在汇编语言的基础上,增加高级语言中控制语句成分(如条件语句,循环语句,函数和过程等)而成的汇编语言,它既保持汇编语言的有效性和灵活性,又充分发挥了高级语言简单、直观、易于编写等优点。条件汇编语言是在汇编语言中引进“条件转移”和“无条件转移”等汇编指示。这将为用户提供一种简便、灵活的剪裁(选择或跳过源程序)的手段。

(曹东启)

huihe

会合(rendezvous) 两任务间的一种同步机制。又称汇合。**Ada语言**中两个任务的相会。为了使两个任务同步和通信,两个任务必须相会,先到达的要等后到达的,两者都到达了,会合就开始。Ada的会合双方是不对称的:一个主动任务,一个被动任务。主动任务通过入口调用发出约请,而被动任务通过接受语句表示接受约请,准备会合。在任务规约中有入口声明任务,从这个入口的意义上说,是被动任务,因为这个入口是为另外的任务准备的,由另外的任务发出对这个入口的调用(一个任务如果调用自己的入口,必然会产生**死锁**)。会合后,两任务独立地继续它们各自的运行。

会合概念最早是由J. Conway在1963年提出的,它结合了同步与消息传递。这种机制后来由C. A. R. Hoare与B. Hansen重新肯定,这是第一种高级同步机制,在Ada语言中已经采用,对于分布式系统来说,这种同步机制比较合适。

参考文献

徐家福. 系统程序设计语言. 北京:科学出版社,

1983 （程虎）

huihua jiechi

会话劫持(session hijacking) 结合网络嗅探、欺骗与报文伪造技术的一种高级攻击技术手段。从广义上说,会话劫持就是在一次正常的通信过程中,攻击者作为中间人参与到其中,或者是在交互会话里修改通信内容或注射额外信息,或者是将双方通信模式暗中改变,从直接联系变成由攻击者中转。狭义上的会话劫持专指TCP会话劫持技术。

会话劫持的基本分类

根据通信模式的不同,会话劫持主要分为两类:中间人攻击会话劫持与注射式会话劫持。

(1) 中间人攻击会话劫持 在中间人攻击模式中,攻击者借助网络欺骗技术,将本来通信双方直接联系的过程,改变为经过攻击者第三方中转的过程。采用中间人攻击模式的具体会话劫持攻击技术有TCP会话劫持、SMB会话劫持、HTTPS中间人攻击等。

TCP会话劫持(TCP Session Hijacking)是最主流的一种会话劫持技术,利用TCP协议缺乏有效身份认证机制的缺陷,通过网络嗅探获取或利用可预测性缺陷猜测出TCP传输序列号,并发送伪造源IP地址和TCP序列号的数据包,劫持通信双方已建立的TCP会话连接,作为中间人对通信双方假冒对方的身份,从而操纵整个通信会话过程。

(2) 注射式会话劫持 注射式会话劫持比中间人攻击实现起来简单一些,它不会改变会话双方的通信流,而是在双方正常的通信流插入恶意数据。属于此类的具体攻击技术包括RST攻击、DNS劫持、路由劫持等。其中最常见的是RST攻击,是一种通过发送假冒通信方IP地址的伪造TCP重置数据包,干扰TCP通信连接造成通信中断的技术方法。

会话劫持技术发展历史

TCP会话劫持是最早被提出的会话劫持技术,起源于1985年R. T. Morris发现的TCP/IP协议栈缺陷与盲攻击技术,在1994年Kevin Mitinick使用会话劫持技术攻击了Tsutomu Shimomura负责安全管理的San Diego超级计算中心,使得会话劫持技术获得了安全社区的广泛认识和关注。随后黑客社区中出现的Juggernaut、Hunt、TTY Watcher等专用TCP会话劫持工具使得该项技术更加流行。此后,黑客社区在SMB、HTTPS、DNS等常用协议中又逐步发现出安全缺陷,并开发出多样化的会话劫持攻击技术。

会话劫持攻击防范技术

针对TCP会话劫持攻击中TCP序列号可预测的安全缺陷,IETF制定了RFC 1948通过随机化初始序列号避免盲攻击,但仍未解决采用中间人攻击模式的会话劫持攻击问题。目前针对会话劫持攻击的防范技术包括:①防御网络嗅探与欺骗技术,避免会话敏感信息泄露;②采用IPsec、VPN等网络层加密机制和SSH等应用层安全协议,避免高层协议敏感信息被嗅探窃取;③采用Kerberos等协议加强通信双方的身份认证机制。

参考文献

Bellovin S. Defending against sequence number attacks. RFC 1948. May 1996 （诸葛建伟）

huituji

绘图机(plotter) 一种能在纸张、薄膜和胶片等记录介质上绘出计算机生成的各种图形或图像的设备。绘图机最早出现于20世纪50年代末,近年发展较快,不断有新型号的绘图机推向市场。目前各种性能规格的绘图机已广泛应用在各行各业和各种科学技术领域之中,成为图形、图像处理系统不可缺少的输出设备。

绘图机的分类 绘图机的类型很多,可以按外形结构、驱动方法、控制系统和介质上图形绘制的方法等进行分类。根据介质上的图形绘制方法的不同可将绘图机分为向量绘图机和点阵绘图机两大类。

(1) 向量绘图机 以线段作为构成图形的基本元素,这些线段是用绘图工具,也就是包括圆珠笔、墨水笔、铅笔等在内的各种笔,以及刻刀、刻针、光笔等在媒体上画出来的,所以又将向量绘图机称为笔式绘图机。向量绘图机在结构上有滚筒和平板两种类型。

(2) 点阵绘图机 以点作为构成图形的基本元素,实际是用点阵图像表示图形,由于图形不是用笔绘制出来的,所以也称为无笔绘图机。

以上两种绘图机都能按计算机的命令绘出各种图形。但是向量绘图机不能接收和处理计算机输出的点阵绘图命令,而点阵绘图机不管计算机输出的是点阵绘图命令还是向量绘图命令,它都用点阵表示出来。

绘图机的工作原理 以下分别阐述滚筒式向量绘图机、平板式向量绘图机以及点阵绘图机的工作原理。

(1) 滚筒式向量绘图机的工作原理 该绘图机的基本结构如图1所示。其工作原理如下:将绘图

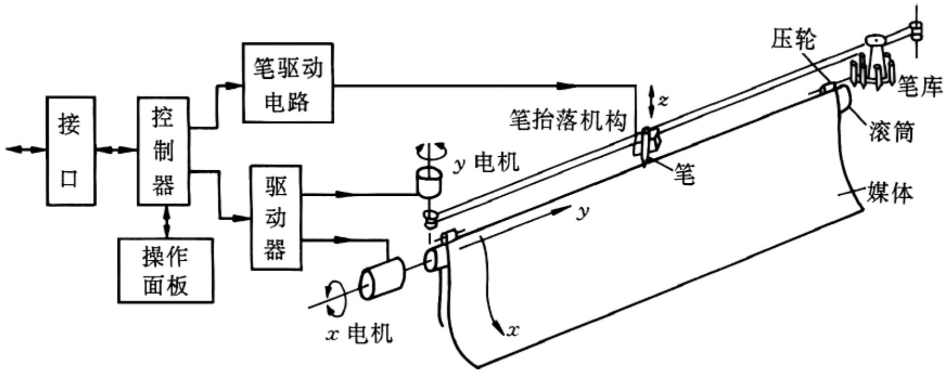


图 1　滚筒式向量绘图机结构示意图

介质用具有很大摩擦力的压轮压在旋转的滚筒上,驱动器驱动一只电机带动这个滚筒滚动时,也就带动绘图媒体在它的 x 轴方向运动,另一只电机带动笔架在 y 轴方向运动;同时由笔驱动电路驱动笔架中的笔抬落机构在 z 轴方向运动,在需要绘图的地方落笔,在不需要绘图的地方抬笔,画出想绘的图形。在绘制不同粗细或不同颜色的线条时,绘图机能自动地到笔库中换笔。一般在笔库中可安装 4 支到 16 支不同粗细或不同颜色的绘图笔供自动选用,所以绘图机能在一幅图内绘出多种宽度和多种颜色的线条。滚筒绘图机中有些型号对介质的长度没有限制,如果采用成卷的介质和相应的输纸机构,在连续绘图时可以不必逐张更换介质。滚筒绘图机的结构简单,速度和加速度高,占地面积小,使用方便,价格也比较低廉,但绘图精度相对低一些。适用于对图形精度要求不太高,但出图速度要求较高的场合。近几年来滚筒式向量绘图机的发展很快,也有不少高精度产品问世。

(2) 平板式向量绘图机的工作原理　该绘图机的基本结构如图 2 所示,其工作原理如下:采用静电吸附或磁性压条等方法将绘图媒体固定在平板上,一只电机推动横梁在 x 轴方向运动,另一只电机推动横梁上的笔架在 y 方向运动,其他绘图笔在 z 轴方向的抬落、换笔等与滚筒式向量绘图机相同。平板式向量绘图机的机械结构较为复杂,由于横梁的重量较重,对控制系统的要求也就较高,速度、加速度也相对较低。平板式向量绘图机可用许多方法将精度做得很高,但价格也随精度要求提高而上升。高精度平板式向量绘图机适用于对图形精度要求较高的场合。

(3) 点阵绘图机的工作原理　该绘图机的基本结构如图 3 所示,由图可见,点阵绘图机结构是滚筒式的,它与滚筒式向量绘图机相比只是将向量绘图机的笔抬落驱动器换成了点阵写入电路,笔架改为点阵写入头。如果点阵绘图机从计算机接收到的是点阵数据,则由控制器直接将这些数据送往点阵写入电路,由写入头写入介质;如果从计算机接收到的是向量绘图命令,就要经过向量点阵变换器将向量变换成点阵数据送入点阵写入电路。向量点阵变换器可以用控制器中的程序实现,但为了提高变换速度,一般采用硬件实现。

点阵绘图机的主要优点是不管图形的复杂程度如何,出图的速度几乎一样,在图形较为复杂时,出图的速度要比向量绘图机快得多。点阵绘图机绘出的图形画面色彩丰富,适合表现三维立体图形的表面涂色、光照等效果以及计算机的图像输出。点阵绘图机的工作噪声很小。

利用静电、喷墨、激光、热转印和发光二极管等



图 2　平板式向量绘图机结构示意图

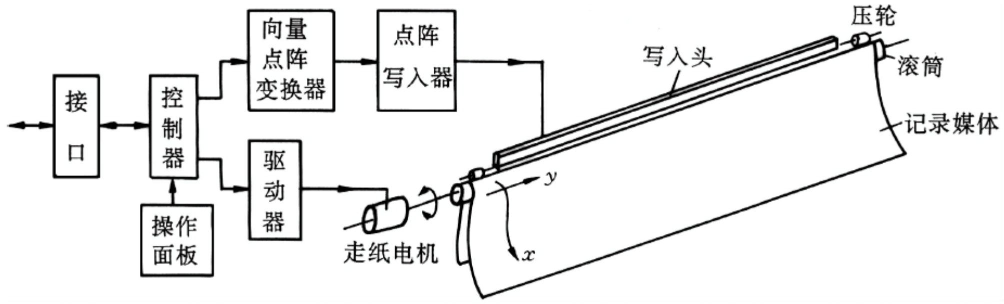


图 3 点阵绘图机结构示意图

各种在介质上写入点阵的技术都可以制成点阵绘图机。

绘图机控制器和接口 不同类型的绘图机中的控制器和接口的基本功能都一样。

绘图机的控制器是以**微处理器**为核心构成的控制系统,在它上面运行一个固化了的控制程序。控制器一方面要接收用户从操作面板键入的命令,另一方面逐条解释从计算机送来的绘图命令,根据这些命令的要求控制绘图机工作,完成绘图任务。从计算机送来的绘图命令中通常包括直接绘图命令和控制命令,前者如绘制线段、曲线、文字、符号和点阵数据等,后者如换笔,改变速度和加速度,改变笔压力等。这些命令的集合就是绘图机的绘图语言(也称为绘图机的命令集)。目前为了解决兼容问题,绘图语言已逐步标准化。

为了能和不同的计算机相连接,绘图机一般都选用某种标准的设备接口,如 RS-232 串行接口。

绘图机的主要性能指标 向量绘图机的主要性能参数定义如下:

(1) 分辨率 绘图工具沿 x 轴或 y 轴运动的最小长度值,或称为绘图机的步距。

(2) 有效绘图幅面 确保绘图工具正确绘图的最大区域,用长 × 宽表示。

(3) 绘图速度和加速度 绘图工具相对绘图介质沿轴向运动的最大速度和加速度。

(4) 绘图精度 由机械系统、控制系统和换笔等因素引起绘制图形的实际尺寸值与理想尺寸值之差,也称定位误差。

(5) 笔数 笔库中可同时安放的笔的支数。

中华人民共和国国家标准 GB 9814—1988《向量绘图机通用技术条件》中对向量绘图机的等级和档次作了规定。

一般绘图机的速度、精度和有效绘图幅面是相互制约的性能指标。例如,绘图精度要求很高时,有效绘图幅面就不能做得很大,同时速度、加速度也不能提得很高。

点阵绘图机性能指标中的分辨率、精度、速度和色彩等的定义和规定都与相应的点阵印刷设备(如针式打印机)基本相同。点阵绘图机的有效绘图幅面、接口和命令集等与向量绘图机的要求和规定基本一样。

向量(笔式)绘图机,特别是光绘图机、刻图机等专用笔式绘图机将会有较快的发展;点阵绘图机由于突破了关键技术,有可能后来居上。绘图机技术发展的趋势是提高绘图机的图形处理能力和绘制能力;增加数据吞吐量和设备的通用性;提高绘图机的精度和色彩表现能力,使图形更加精确,色彩更加丰富;提高出图速度,从而提高工作效率;绘图命令集进一步扩展,并逐步标准化,从而改善软件的兼容性;扩大随机存取存储器(RAM)缓冲或增加磁盘缓冲存储,增强脱机重绘的功能;扩大自诊断功能和人机交互操作能力。此外,笔式绘图机还要进一步降低各种因素引起的噪声。

参考文献

1. 郭平欣,姚锡珊,虞浦帆. 电子计算机外部设备原理. 北京:国防工业出版社,1986

2. 张江陵,季国钧,等. 电子计算机外部设备设计原理. 武汉:华中理工大学出版社,1989

(周利华)

huizhi jishu

绘制技术(rendering techniques) 将场景的数字化表示模型转变为图像的技术,根据场景表示模型和输出图像的不同,需采用不同的技术来实现。

绘制技术研究始于 20 世纪 60 年代,1968 年 IBM 公司研究中心的 Arthur Appel 最早提出了光线投射(ray casting)方法。Bouknight 在 1970 年提出了基于扫描线的绘制方法,Catmull 于 1974 年首次提

出了 Z-buffer 消隐算法以及纹理映射方法。光线跟踪(ray tracing)方法于 1980 年由 Whitted 提出。80 年代开始,人们开始致力于研究如何模拟光能在景物间的传播,先后提出了辐射度(1984)、光子跟踪(1995)、预计算辐射度(2002)等方法。90 年代中期出现了基于图像的绘制技术,代表性的成果有 QuickTime VR(1995)、Plenoptic modeling(1995)、Light Field(1996)、Lumigraph(1996)等,之后基于图像的绘制技术得到了快速发展。2000 年之后,基于点的绘制技术随着 QSplat 方法(2000)的提出而逐渐得到研究人员的重视。随着计算能力的大幅提升,以及**计算机图形学**和**计算机视觉**的相互融合,基于视频的绘制技术得到了快速发展。

真实感图形绘制技术以表现场景真实的光照效果为目的;如果需要在画面上表现某种艺术效果,则需要采用**非真实感图形绘制**技术。

根据产生图像的速率,可分为**实时绘制**技术和离线绘制技术。前者每秒钟至少应产生 24 帧以上的图像,常用于视频游戏、实时仿真和虚拟现实等;后者则没有图像产生速率的限制,常用于高度真实感的数字电影特效和动画制作等。

根据输入场景的数字化表示模型的类型,可分为基于多边形模型的绘制技术、基于高次曲面的绘制技术、基于点的绘制技术、**基于图像的绘制**技术、基于视频的绘制技术、体绘制技术以及基于混合模型的绘制技术。

基于多边形模型的绘制技术的应用最为广泛,因为其所采用的场景多边形表示模型最为常见,表达能力最强,也最通用。

基于高次曲面的绘制技术则采用高次曲面作为场景的数字化表示模型。多边形表示模型实际上可视为 1 次曲面。基于高次曲面的绘制技术往往将高次曲面全局性的或是局部性的转化为多边形表示。

在基于点的绘制技术中,景物由其表面上大量稠密的三维离散采样点集合表示。由于不存储采样点之间的拓扑联系,数据结构简单,适合于表示具有复杂表面细节的场景。

在基于图像的绘制技术中,场景由单张或若干张图像来表示。基于图像的绘制技术实质上是基于已有的场景图像集合生成一张新图像,进行这种转换的原因大多因调整了观察参数,或是改变了光照条件,或是修改了原先照片里的一些内容。该技术的特点是绘制速度快,真实感高,并且场景表示可通过实拍照片获得。

基于视频的绘制技术是在基于图像的绘制技术上发展起来的。其中场景是由一个或多个视频来表示,一个视频就是一段图像序列,其中的图像在时间上具有连续性。基于视频的绘制就是充分利用这种连续性来改进传统的基于图像的绘制方法,产生新的绘制效果。

体绘制的对象是基于体表示的三维物体或空间。所谓的三维体表示,即不仅描述体的表面形状,而且描述在体内部各点处的信息。常见的三维体表示为正规/非正规四面体、正规/非正规六面体模型。

基于混合模型的绘制技术是综合采用多种场景表示模型来描述场景中的不同物体或同一物体的不同细节层次。根据不同的混合方式,形成了不同的绘制方法。

从绘制的过程来看,一般会分为软件处理阶段、几何处理阶段和像素处理阶段。在软件处理阶段,一般完成场景遍历、层次细节计算或选取、可见性剔除等操作,最终生成绘制元素集合,如三角面片、纹理片段等。在几何处理阶段,主要进行顶点观察坐标系变换、顶点光照明计算、投影变换、裁剪等,最终输出三角片面的顶点信息,包含几何坐标、颜色、纹理坐标等。在像素处理阶段,其核心是对三角面片进行光栅化。在光栅化的同时可对每个像素进行纹理映射和颜色计算。软件处理阶段一般在主机中由 CPU 中完成,在最新的可编程图形硬件支持下可部分在 geometry shader 中由 GPU 完成;几何处理阶段在 vertex shader 中由 GPU 完成;像素处理阶段则在 pixel shader 中由 GPU 完成。

参考文献

1. Heung-Yeung Shum, Shing-Chow Chan, Sing Bing Kang. Image-based rendering. Springer Science + Business Media, LLC. 2007

2. Ian Stephenson (ed.) Production rendering, design and implementation. London: Springer. 2010

3. Marcus A. Magnor, Marcus Andreas Magnor. Video-based rendering. Natick, MA: A K Peters, 2005

4. Jusub Kim. Volume Rendering: Out-of-Core Algorithms and Parallel Rendering. VDM Verlag, 2008

5. Alan H. Watt, Mark Watt. Advanced animation and rendering techniques: theory and practice. Addison-Wesley. 1992

(鲍虎军 华炜)

huizhi yinqing

绘制引擎(rendering engine) 一个支持图形

绘制的软件环境。由离线的工具集和运行的支持环境两部分构成。其中离线的工具集中主要包含从第三方建模软件进行数据导出的工具，以及对场景模型(scene model)进行某些预处理的工具(例如简化工具、可见性预计算工具、光照预计算工具、数据压缩工具等)。更高级的绘制引擎还会包含一些诸如特效生成器在内的开发工具。运行的支持环境的最主要部件有场景模型、场景模型管理器(scene model management)、场景图遍历(scene graph traverse)和绘制器(render)：

(1) 场景模型是虚拟世界在网络空间的数字描述形式。在大多数绘制引擎中，场景模型采用有向无环图的形式进行组织，称为场景图(scene graph)，场景图中的节点表示虚拟世界中的实体(绘制实体和光源实体是最常见最重要的实体)，场景图中的弧表示实体之间的关系(集合关系和引用关系是最常见的关系)。

(2) 场景模型管理器是绘制引擎中管理场景模型的一个模块，例如对场景图节点的操作(创建、修改、引用、复制、重命名、搜索、删除等)。场景模型管理器功能一般包括：场景模型节点的生命周期管理、节点的属性管理、节点之间的关系管理、空间结构管理、场景查询、场景模型的载入和保存。

(3) 场景图遍历模块负责遍历场景图的每一个节点，开发人员可以定义遍历过程中对节点的操作，最常见的有动画更新、可见性判断和 LOD 层次选择等。

(4) 绘制器是对各种输入绘制队列进行绘制的模块，其有一个基本的框架，以协调、安排和管理渲染队列和渲染过程，便于增加新的绘制特效和新节点的绘制，一般包括纹理映射、着色器管理、阴影绘制模块、特效绘制模块等。

应用程序主要是通过调用场景管理器和场景树遍历这两个模块中的相应函数来使用绘制引擎。

除了以上的核心组成部分外，绘制引擎为了便于应用开发者使用，还会提供一系列辅助模块。比较常见的有：①观察器控制：对虚拟场景中观察者的位置、朝向、视域范围、运动速度/角速度、运动路线、图像传感器特性(如传感器平台抖动、噪声模拟、运动模糊等)进行控制。②视觉特效：模拟地面/水面爆炸、爆炸碎片、武器开火闪光、飞行导弹尾迹、螺旋桨旋转、旋翼下降气流、燃烧烟雾等效果。③显示环境配置：支持包括 CAVE、Power wall 等类型的显示设备，用户可以配置投影显示面数量、排列方式、立体显示双眼间距。同时支持非线形失真校正，支持圆柱、平面投影幕以及多投影幕边缘融合。

参考文献

Bao Hujun, Hua Wei. Real-time graphics rendering engine. Hangzhou: Zhejiang University Press, Heidelberg: Springer, 2010

(鲍虎军 华炜)

hunhe guangxian tongzhou dianlan

混合光纤同轴电缆(hybrid fiber coaxial cable, HFC) 基于电缆电视(CATV)网的光纤**光缆**和**同轴电缆**混合网的**宽带接入体系结构**。它是以同轴电缆网络为最终接入部分的宽带网络系统。

HFC 的网络结构是在头端到光节点的光网络单元之间通过光纤光缆传输光波信号，光节点到用户家之间则利用原有的同轴电缆分配网来传输和分配用户信息。它既保留了传统的模拟传输方式，又实现了上、下行双向和模拟数字混合传输，能同时提供电话、模拟视频、数字视频和交互式业务。

HFC 宽带接入网涉及的关键技术：

(1) 上行通道噪声与干扰的抑制技术 由于 HFC 的同轴电缆部分是一点到多点的树状结构，反向通道的噪声是各支路放大器的级联噪声和各支路间噪声的叠加，形成了噪声的漏斗效应。反向通道噪声源有四类，即窄带短波噪声，频带为 5 ~ 30 MHz；冲击噪声，频带为 60 ~ 2 MHz；由设备的非线性引起的共模失真以及用户在使用各种电器时产生的频带在 30 MHz 以下的本地干扰噪声。抑制噪声的措施有以下几点：①选用屏蔽特性好的电缆，以消除从空间耦合进入电缆的噪声干扰；②选用连接特性好，无泄漏电磁波的电缆连接器，以消除因连接泄漏而引入的噪声；③室内电缆端口用匹配器连接，防止从端口接收和发射干扰；④及时更换接触不良或锈蚀的电缆及接头；⑤选择经严格培训合格的人员精心操作安装。

(2) 上行通道多址接入技术 在 HFC 上行通道中常用三种多址接入技术，即时分多址接入(TDMA)、频分多址接入(FDMA)和码分多址接入(CDMA)。**时分多址**是将用户的数据流分配到特定的时隙；**频分多址**是每个用户的数据流都有一个载频；**码分多址**是为每个用户分配一个特定的扩展序列。每一种多址接入技术对 CATV 上行通道的干扰承受能力是不同的。对于多址接入的干扰(MAI)，TDMA 和 FDMA 由于信号排列在时隙或频率隙内，相邻时隙或频率隙影响可以忽略不计，而 CDMA 的容

量则受 MAI 的限制。

（3）下行通道的数模混合传输技术　HFC 的 CATV 网，下行不仅传送多路模拟视频信号，还传送多路数字电视，因此，下行信息是一个模拟、数字混合传输的信道。设计时，既要保证模拟信道指标要求，又要满足数字信道误码率要求。模拟信号和数字信号的叠加，会产生尖峰脉冲。数字信号的引入要保证不影响模拟视频信号，总的光调制指数保持不变。模拟视频信号的限幅失真会使数字信号的误码率大大增加，采用合适的编码方式、调制方式和纠错技术，可以减少限幅噪声对数字信号的干扰，保证数字信号的误码率达到要求。

HFC 可以充分利用现有的 CATV 网宽频的特点，不需重新敷设配线网。除了提供有线电视节目外，还可为用户提供电信业务和宽带交互式多媒体业务。HFC 接入的标准主要来自有线电视实验室（CableLabs）发布的电缆业务数据接口规范 DOCSIS（date over cable service interface specifications），包括 DOCSIS 1.0，2.0 和 3.0。目前 DOCSIS 的标准基本完善，已有许多基于 DOCSIS 的 HFC 综合接入设备进入商用阶段。

参考文献

1. 胡道元. 智能建筑计算机网络工程. 北京：清华大学出版社，2002

2. 刘伟. 宽带综合接入技术. 北京：科学出版社，2007

（程时端）

hunhe jisuan moxing

混合计算模型（hybrid computational models）　计算机与其所控制的物理部件构成的，具有既随时间连续变化的变量又受事件驱动的离散变量的系统的数学模型。

混合系统的设计涉及控制理论和计算机科学，20 世纪 90 年代以来引起了计算机科学界很大关注。很多混合系统要求绝对安全，如自动导航系统、核电站监测系统等。绝对安全系统的设计是计算机软件科学的重大课题。当前的软件产品耗资巨大，但多数无法避免差错。这样的软件不能用于绝对安全系统。计算机软件科学界提出使用严格的形式化方法来设计该类软件。

迄今，形式化方法所处理的对象都是离散变量（或已离散化的变量），所用的语言及演算亦都是基于离散数学的公理化系统。而混合系统设计的形式化方法不可回避连续数学；控制理论使用的是微分方程刻画连续变量的变化规律。混合系统设计的形式化方法也不能回避离散的事件，这些事件驱动系统的变化。混合系统设计的形式化方法需要一种计算模型，它同时支持连续变量和离散事件耦合系统的计算。

近几年在已有计算理论的基础上，已陆续发展了多种混合计算模型。大体上可分为逻辑型、程序设计型和自动机型三类。

逻辑型混合计算模型的主要思想基于时态逻辑，引入时段和切变的概念。时段可用来刻画系统在一个时间区间上的连续变化，而切变则表示事件的发生（离散变量的变化）。在单个时段上，借用连续数学（微分方程理论）推导系统的行为；而在相邻时段间，则用时态逻辑中切变算子的规则，推导系统行为的转化。逻辑型计算模型中的时段演算，已引起该领域同行的广泛重视。该演算是由周巢尘、C. A. R. Hoare 和 A. P. Ravn 所建立。

程序设计型混合计算模型是将传统的程序设计语言加以推广以容纳连续变量。推广后的程序语言可用来描述混合系统的行为。而其中的控制部分可逐步求精，变换成传统的可在计算机上执行的软件，从而生成数值控制系统。通信顺序进程 CSP，已推广为混合通信顺序进程。在这个程序语言中，有一种特殊的语句称为连续构件，它可表示一个具体给定初值的微分方程；而原有的通信语句可用来表达事件的起源和发生；程序语言中的顺序算子、条件算子等用来刻画连续构件和通信间的耦合关系。

自动机早已用于各种模拟计算系统，计算机本身亦可看作一个庞大的有限状态自动机：一个状态表示计算机中各存储器和寄存器一种取值，而计算机的操作导致计算机由一个状态转移至另一个状态。如果将自动机的状态看作是在一组微分方程控制下，一组连续变量的连续变化过程，则将状态的转移视作事件的驱动。这种推广后的自动机称作混合自动机，可用来描述和计算混合系统的行为。

总之，混合计算模型还在发展完善之中。

参考文献

1. Zhou Chaochen. A calculus of durations. Information Processing Letters，1991，40(5)

2. He Jifeng. From CSP to hybrid systems，A classical mind. Prentice-Hall，1994

3. Alur R，et al. Hybrid automata：an algorithmic approach to the specification and verification of hybrid

systems. In: hybrid systems, LNCS 736, Springer-Verlag, 1993: 209-229 (周巢尘 李晓山)

hunhe xianshi

混合现实(mixed reality, MR) 通过虚拟环境与现实环境的融合,在虚拟世界、现实世界和用户之间构成一个相互作用的回路,使得用户能更好地感知现实世界和虚拟世界的方法和技术。混合现实是**虚拟现实**技术的发展。目前有两种相互作用模式,一是将虚拟对象或环境作用到现实环境中来增强用户对现实世界的感知,二是将现实世界中的信息实时地作用到虚拟环境中来提高虚拟环境的准确性。因此,混合现实系统具有以下3个特征:①虚实对象或环境具有一致的时空关系(即在时间和空间上是准确整合的);②具有实时人机交互功能;③虚实对象或环境信息在用户感知上是融合一体的。这里,"虚拟对象"指由计算机生成或模拟出来的模型,而"现实对象"则泛指由实物本身及采集自现实世界的信息或其重建模型。在混合现实系统中用户不再与其所处的真实世界相隔离,而是处于虚实相融合的混合环境中。

混合现实的概念由Paul Milgram和Fumio Kishino于1994年提出。他们根据虚拟环境与现实环境的混合程度及作用形式将虚实混合环境分为四类,即虚拟环境、增强虚拟型环境、增强现实型环境和现实环境。其中,现实环境即为现实世界,虚拟环境为传统虚拟现实技术所模拟的虚拟世界,增强虚拟型环境由现实对象或环境叠加融合到虚拟环境中而成,而增强现实型环境则由虚拟对象或环境叠加融合到现实环境中而成。因此,混合现实技术亦派生出相应的**增强虚拟**(augmented virtuality, AV)和**增强现实**(augmented reality, AR)两种技术。这两种技术本质上没有区别,只是叠加融合时的主环境(虚拟环境或现实环境)不同而已。

除了自然交互以外,混合现实技术的核心是虚拟环境与现实环境之间的无缝融合,包括现实对象或环境的信息的实时感知、虚实对象或环境的高精度空间注册、虚实对象或环境的逼真绘制和融合呈现等。这种融合使得混合现实系统在用户交互操作时,虚实对象或环境能够始终保持一致的时空关系。因此,混合现实技术极大依赖于传感器、实时的信息获取、处理和融合显示等硬软件技术。例如,可以利用虚拟现实技术生成虚拟景物的画面或信息,直接投影到现实场景中,实现自然的虚实融合;也可以采用数字摄像机来实时获取现实场景,并通过三维计算视觉技术从获取视频中恢复场景的几何结构来实现虚实环境的空间注册,进而利用场景的光照信息绘制出虚拟对象或环境,最后将虚实画面融合起来输出到显示器中。

不同的混合现实系统,通常采用不同的融合显示方式。目前,混合现实系统的实现方法主要有5种:①基于光学的透视式头盔显示器实现方法;②基于光学的台式图形监视器实现方法;③基于视频的透视式头盔显示器实现方法;④基于视频的台式图形监视器实现方法;⑤基于实物表面直接投影的实现方法。特别地,近年发展起来的微投影显示技术使得虚拟景物可移动投影融合到实物表面上,为移动式混合现实的应用奠定了基础。

经过多年的发展,混合现实技术的应用日益广泛。除了传统的交互模拟仿真或训练等应用外,混合现实技术还可以将从现实世界获取的信息实时或远程地融合到相应的虚拟环境中,从而在现实环境、虚拟环境和用户之间建立起相互控制和反馈的通路,实现对现实复杂系统(如远程无人装置和大型装备或生产流水线等)的在线运行分析和操控。这一技术特点使得混合现实技术较之虚拟现实技术具有更广泛的应用背景。一旦信息获取、传输、处理、融合和呈现等关键技术取得更大的突破,其应用前景是非常美好的。

参考文献

1. Milgram P, Kishino F. A taxonomy of mixed reality visual displays. IEICE Trans on Information Systems, 1994, E77-D, (12)

2. Tamura H, Yamamoto H, Katayama A. Mixed reality: future dreams seen at the border between real and virtual worlds. IEEE Computer Graphics and Applications, 2001, 21(6): 64-70 (鲍虎军)

hunhe zidongji

混合自动机(hybrid automaton) 一种描述混合系统的计算模型。它是有限状态自动机的推广,可用来描述和计算具有连续和离散变量的混合系统的行为。在混合自动机中,状态被看作是在一组微分方程控制下,一组连续变量的连续变化过程;而将状态的转移视作事件的驱动。

混合自动机最早是由美国R. Alur, T. A. Hensinger和Z. Mazna等在20世纪90年代以后提出,并给出了一些有关理论结果,如可达性问题的不可判

定性，以及受限的线性混合自动机的一些性质。下面我们用例子简要介绍一下混合自动机。

图 1 表示一个线性混合自动机，且只有一个数据变量 x。它所描述的系统拥有两个状态位置 l_1 和 l_2。在位置 l_1 中，x 的数值减少的速度为 1，即 $x' = -1$。从位置 l_1 到位置 l_2 的迁移，在 $5 \leqslant x \leqslant 6$ 的任何时刻都可能发生。当迁移发生时，x 的数值即刻减 1，进入位置 l_2。一旦进入位置 l_2，x 将以 2 的速度增加，即 $x' = 2$。只要 x 的数值一达到 10，迁移发生，进入位置 l_1。事实上因为位置 l_2 具有不变式 $x < 10$，$x = 10$ 时，系统已经在位置 l_1 了。

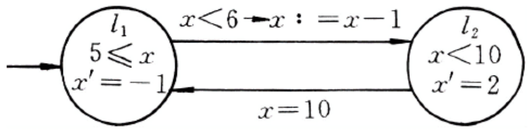


图 1

用混合自动机描述混合系统的优点在于直观，其缺点是不易进行推导和求精。

参考文献

Alur R et al. Hybrid Automata: an algorithmic approach to the specification and verification of hybrid systems. In: Hybrid Systems, LNCS 736, Springer-Verlag, 1993: 209-229

（周巢尘　李晓山）

huoqu guocheng

获取过程（acquisition process）　获取者获得一个系统或软件产品的一系列活动。它从确定获取该系统或软件产品的需求开始，经过招标准备、合同准备、谈判及修改、对供应方的监督等活动，直至验收完成方告结束。获得该系统或软件产品的机构可就部分或全部获取活动与某机构签订合同，后者根据获取过程开展相应的活动。二者皆可称作获取者。

获取者可根据项目具体情况对这些活动进行剪裁和组织（参见**剪裁过程**）。

获取过程主要包括以下各项活动：

（1）开始　获取者确定要取得的一个系统或软件产品，并对其需求进行定义，选择获取方式，制定计划、验收策略、准则和条件。它有以下具体任务：①系统需求定义。按照**开发过程**中有关活动内容来进行。需求定义中最好要包括安全性、保密性和其他关键需求以及要求设计、测试需遵循的有关标准和规程。如果由供应者完成需求定义，则结果必须要经获取者的同意和认可。②获取方式的选择。获取者应根据项目的具体任务和风险大小来选取获取方式，获取方式一般有以下几种：(a) 购买现货产品；(b) 自行开发；(c) 通过合同委托开发；(d) 上述三种方式的结合；(e) 改进现有的软件产品。当选用方式(a) 时，该现货产品必须要具备能满足需求、有必备的文档、能达到所有权和使用权要求，并有未来的产品支持计划等条件。如有多个现货产品可供选用时，最好还要按有关评价选择标准等作为准则，进行优选。③以文档形式制定获取计划。计划中要包括系统需求、使用定义、执行合同的类型、所涉及机构的责任、将使用的支持概念、预计风险以及解决这些风险的办法等内容。④确定验收策略和准则。

（2）招标准备　①根据开始活动中选用的获取方式编写一份系统获取需求的文档，即标书，其中要包括系统需求、项目描述、投标者须知、软件产品清单、子合同或合同条款、技术制约（如目标环境）等内容。②根据项目决定采用哪些过程和活动，并对它们进行相应剪裁，还要特别指明可应用的**支持过程**和它们的执行机构，以让供应者在投标书中确定各个特定支持过程的执行方法。③确定供应者在**供应过程**中将接受评审和审计的合同阶段目标，并作为对他们进行监督的一部分（参见**支持过程**）。④把获取需求文档交给实施获取活动的有关机构。

（3）合同的准备和修改　①确定选择供应者的方法和步骤，其中要包括对投标书的评价准则和其符合需求的程度。②对供应者的投标书、能力及其他需要考虑的因素进行综合评估，并以此为基础选择一个合适的供应者。③按项目剪裁各项活动及内容，并将最终决定包含于合同内容中。④准备一份合同，并就此与供应方商议和谈判，商议时要提出系统的需求、成本和交付软件产品的日程，合同中还要涉及可供使用的现货产品的产权、使用权和所有权等条款。⑤合同一旦生效，获取者将通过与供应者谈判来控制合同的修改，同时要了解该修改对项目计划、成本、质量和日程等的影响。

（4）监督供应者　获取者将进行支持过程中的联合评审和审计过程，对供应者实施监督，必要时还要采纳支持过程中的验证和确认过程（参见**支持过程**）。在整个获取过程中，获取者要与供应者密切合作以便及时提供全部必要的信息和解决所有有待共同解决的问题。

（5）验收完成　根据原先已确定的验收策略、准则和条件为验收作好全部必要的准备，诸如测试

用例、测试数据、测试规程和环境准备等,然后对交付的软件产品进行验收评审和测试,当产品符合所有的验收要求后,便予以接受。验收后,需要时还可对已交付的软件产品进行配置管理(参见**支持过程**)。

参考文献

1. IEEE Standard for Developing Software Life Cycle Processes —IEEE Std 1074—1991

2. ISO /IEC 12207:1995 Information Technology—Software—Part 1: Software Life-Cycle Process. 1995

(陈森芬)

Huoen luoji

霍恩逻辑(Horn logic) **一阶逻辑**的子部分。原子公式称为正文字,原子公式的否定称为负文字,它们统称为文字。例如,$P(a)$,$P(f(x))$是正文字,$\neg Q(a,f(x))$是负文字。形式为$L_1 \vee \cdots \vee L_m$的公式称为子句,其中$L_1,\cdots,L_m$是文字。用□表示不包含任何文字的空子句。它是一个永假式,即总是解释为假命题的公式。至多包含一个正文字的子句称为霍恩子句,是逻辑学家 A. Horn 于 1951 年首先研究的。例如,$P(a)$,$\neg P(a) \vee \neg Q(a,f(x))$,$P(a) \vee \neg P(x)$都是霍恩子句,而$P(a) \vee P(f(a))$却不是霍恩子句。

霍恩逻辑是由霍恩子句组成的一阶逻辑的子部分。它是逻辑程序设计的理论基础。在逻辑程序设计中采用了一种特殊的子句记号,用$A_1,A_2,\cdots,A_k \leftarrow B_1,B_2,\cdots,B_n$表示子句$A_1 \vee \cdots \vee A_k \vee \neg B_1 \vee \cdots \vee \neg B_n$。霍恩子句共分以下四种形式:①$A \leftarrow B_1,\cdots,B_n$;②$A \leftarrow$;③$\leftarrow B_1,\cdots,B_n$;④空子句□。其中①称为过程,②称为事实,③称为目标。

霍恩子句可以表达知识库和对知识库的询问。例如,已知"张三在哪儿,他的狗就在哪儿"和"张三在火车上",询问:"张三的狗是否也在火车上?"用个体常元 Zhang,train,dog 分别表示张三、火车、张三的狗,用$AT(x,y)$表示x在y上,则"张三在哪儿,他的狗就在哪儿"表示为过程$AT(\mathrm{dog},x) \leftarrow AT(\mathrm{Zhang},x)$,"张三在火车上?"表示为事实$AT(\mathrm{Zhang},\mathrm{train}) \leftarrow$,问题"张三的狗是否也在火车上?"表示为目标$\leftarrow AT(\mathrm{dog},\mathrm{train})$。因为上述三个子句的集合没有模型(即没有使它们都有效的结构),因此问题的答案是肯定的,即张三的狗确实也在火车上。

Prolog 语言是以霍恩逻辑为基础的高级程序设计语言。一个 Prolog 程序实际上就是一个由事实和过程组成的霍恩子句集,询问就是一个目标。运行一个 Prolog 程序就是通过计算机判断由程序和目标组成的霍恩子句集是否有模型。

事实上,要判断一个霍恩子句集是否有模型,不必考虑所有的结构,只需考虑一类特殊的结构,即埃尔布朗结构就够了。有模型的霍恩子句集必有埃尔布朗模型。设S是一个霍恩子句集,用S中出现的个体常元和函数符号构造出来的所有无变元项称为S的埃尔布朗域(如果S中不出现个体常元,就增加一个个体常元a)。例如,设S由以下霍恩子句组成:

$$P(x) \leftarrow Q(f(x),g(x))$$

$$R(y) \leftarrow$$

则S的埃尔布朗域为$\{a,f(a),g(a),f(f(a)),f(g(a)),\cdots\}$。满足以下两条件的结构称为埃尔布朗结构:①论域是埃尔布朗域;②每个无变元项解释为自己。

如果两个公式集有相同的模型,则称它们是等价的。一个公式集等价于一个霍恩子句集的充分必要条件是:①该公式集的模型的子结构仍是它的模型;②该公式集的模型的直积仍是它的模型。

由前提$A_1,\cdots,A_n$能推出结论B当且仅当公式$(A_1 \wedge \cdots \wedge A_n) \rightarrow B$是逻辑有效式。因此,研究判定公式的逻辑有效性的算法在人工智能中占有十分重要的地位。一闭公式A是逻辑有效式当且仅当$\neg A$是不可满足式,即没有模型的公式。对于每个闭公式A,可找到一个有穷子句集S,使得A不可满足当且仅当S不可满足。因此,判定公式的逻辑有效性的问题可以归约为判定有穷子句集的不可满足性问题。用归结法证明有穷子句集不可满足的过程可以表述为一组霍恩子句的反驳。因此,对于任何有穷子句集S,可找到一组霍恩子句集S',使得S不可满足当且仅当S'不可满足。

参考文献

1. Kowalski R. Logic for Problem solving. New York:North Holland,1979

2. Lloyd J W. Foundations of logic programming. Berlin:Springer-Verlag,1984

(何自强)

J

jidashi dayinji

击打式打印机(impact printer)　利用机械击打动作使字模隔着色带在纸上印出字来的设备。它是最早研制成功的计算机印字设备。

击打式印字设备分为整字形击打、点阵击打、点阵串行击打、点阵并行击打等4类。

(1)整字形打印设备　基于机电原理,利用电磁铁驱动字锤,隔着纸和色带向选定的字模上击打印字。不同字符的完整字形的字模每击打一次印出一完整字形,字模安置在各种形态的载体上,靠字模载体的机械运动来变换处于字锤击打位置的字符。有的字模本身就起着字锤的作用,隔着色带与纸向平台或圆柱上击打印字。整字形击打印字的优点是印字美观自然,其缺点是噪声大、印字速率低、字符种类少、字体更换不便或不可能、机械磨损问题突出。现在已基本不用了。

(2)点阵击打式印字设备　利用多根针经色带在纸上多次击打印出点阵字符的印字设备。当点阵字形构成的点较少时,虽字形可辨,但不清晰美观。随着技术的进步,打印头可以做到48针,点阵字形质量显著提高,几乎可与整字模印出的质量媲美。

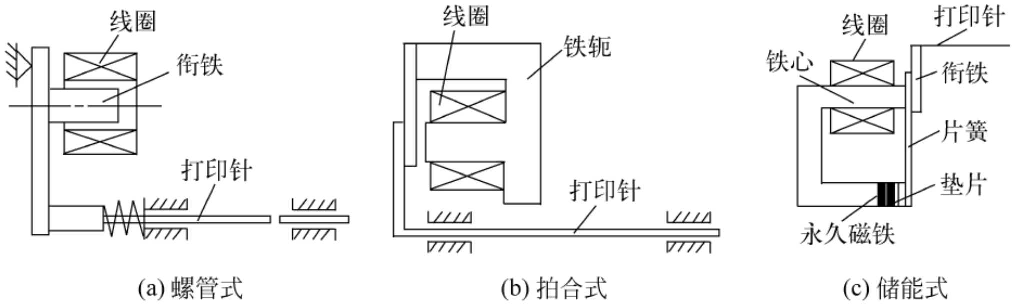
(3)点阵串行击打印字设备　简称为针式打印机,俗称打印机。它的打印头是击打式打印机的成字部件。不同的成字机理,不同的打印头型式,就形成了各种类型的打印机。我国广泛应用的针式打印机的打印头有9,14,24或48根针。针的直径为0.2~0.3 mm。

打印头的基本工作原理是采用一定的动力源驱动打印针撞击打印媒体(色带与打印纸)获得印点点阵而形成所需的文字、数字、符号或图像。

按驱动打印针动力源的不同形式,针式打印头有电磁式与压电式两类。

电磁式打印头利用电磁铁驱动打印针而获得印点。按电磁铁的类型,电磁式针式打印头分为螺管式、拍合式与储能式三种,分别如图1(a)、(b)、(c)所示。螺管式与拍合式同属吸合式,即线圈通电后,衔铁被吸合而驱动打印针撞击打印媒体。储能式打印头则利用永久磁铁的磁场与线圈通电时产生的反向磁场的相互作用使永磁吸力减小,吸力小于簧片弹性恢复力时,簧片释放使打印针撞击打印媒体而获得印点。螺管式打印头虽然稳定性好,噪声较低,但高速性差,故通常用于低档的9针打印机中。拍合式打印头高速性较螺管式的好,且其经济性、可靠性等均属优良,在24针打印机中应用很广。储能式打印头驱动电流小,功耗低,工作频率高,且可靠性高,易于小型化,其高速性、稳定性、低噪声等性能均较上述两者好。因此,在高档、高速、高可靠打印机中有广泛的应用。

压电式打印头的工作原理是利用压电陶瓷材料的逆压电效应(压电晶体在外电场作用下成比例地产生几何形变的现象)驱动打印针撞击打印媒体而获得印点。压电式打印头有两种形式:纵向效应型与横向效应型。纵向效应型采用较多,它利用压电材料平行于外电场方向的伸缩变形以驱动打印针。压电材料是三元系钙钛矿型的铌镍酸铅系压电陶瓷(如NEPEC-10)。由于压电材料的压电常数极小(NEPEC-10的仅约为6.35×10^{-10} m/V),因此,要



(a) 螺管式　(b) 拍合式　(c) 储能式

图1　电磁式针式打印头

使打印针得到约 0.4 mm 的位移量，必须加设扩大机构和采用多层薄片叠层的压电元件结构。

压电式打印头的打印速度高（达 140 汉字每秒），功耗小（约为电磁式的 1/2），体积小，重量轻，便于小型化，且噪声小，价格低，并有很高的寿命（达 10^{11} 次）。

打印机印字的过程是：印字头沿纸面自左向右按步移动，每步一个点距。驱动印针的电磁铁是受字符图形的点阵内容控制的。根据字符点阵的需要，一列印针分别击打或不击打。每完成一列点的打印，印字头右移一个点距。如此一列一列地打印，当印字头横向扫描至右端时，完成一行字符的打印。然后，印字头返回到起始位置，纸前进一个行距。有些打印机具有双向打印的功能，利用印字头的返回过程，在微型计算机的控制下，自右向左逐列打印，以便提高打印速度。

（4）点阵并行击打印字设备　采用多针横行排列的梳状印字头，通常用多个打印头（如 12 个打印头）并排击打。（2003 年）汉字打印速度达到每分钟 450 行，针寿命高达 10 亿次以上。它应用于大批量高速持续打印的场合，如系统的数据备份。

打印机的优点是可使用普通纸，并可同时复印数份。打印机的机构简单可靠，印字速度较快，字型变化灵活，耗材成本低廉。但其噪声大，打印质量差，逐渐地将被打印质量高、噪声低的喷墨印刷机和激光印刷机取代。尽管如此，但由于击打式票据打印机能够在多层纸和厚质材料（如很厚的证券卡或银行存折）上打印，因此票据打印机已广泛地用于处理商务发票、银行存折、报告单据、柜员收据、运输提单以及打印火车、公路、航空、轮船客票等方面。

参考文献

陈其昌，谢仕聘．汉字打印机设计原理．武汉：华中理工大学出版社，1995

（谢仕聘）

jiqi fanyi

机器翻译（machine translation，MT）　用计算机将一种自然语言（源语言）翻译成另一种自然语言（目标语言）的理论、方法和技术，又称**自动翻译**（automatic translation）。属于计算语言学和自然语言处理的研究范畴。

发展简史

早在电子计算机出现以前，人类就曾幻想过实现从一种语言到另一种语言的自动翻译。

1946 年，第一台电子计算机问世以后，一些学者就想到可以用电子计算机来进行语言的翻译。1954 年，美国乔治敦大学在国际商用机器公司（IBM）的协同下，用 IBM-701 计算机，进行了世界上第一次机器翻译试验。随后，全世界范围内出现了机器翻译的研究热潮。中国在 1956 年制定的《1956—1967 年科学技术发展远景规划纲要》明确把自动翻译作为第 41 项任务“计算技术的建立”的主要目标。1959 年，中国科学院语言研究所与计算技术研究所合作的中国第一个机器翻译系统诞生。这一时期的机器翻译系统所采用的方法通常被称为直接机器翻译（direct machine translation）方法。

1966 年美国发表了题为“语言与机器：翻译中的计算机与语言学”的报告（通常称为 ALPAC 报告）。它认为“一般科技文章的机器翻译是不存在的，而且在最近的将来也不会有”，并指出“在目前给机器翻译以大力支持还没有多少理由”。这使得对机器翻译研究的投入急剧减少，导致机器翻译研究陷入了十余年的低潮期。

从 20 世纪 70 年代后期开始，由于计算机技术的发展、**人工智能**和语言学理论的进步，特别是由于社会日益信息化对自动翻译的需求，在世界范围内，机器翻译研究与实用系统的开发再次得到蓬勃发展。这一时期研究者们提出了基于转换（transfer-based）的机器翻译思想，并开始普遍采用**基于规则的机器翻译**（rule-based machine translation，RBMT）方法。同时市场上也出现了一批商品化的机器翻译系统。中国的第一个商品化英汉机器翻译系统也于 1988 年发布。这一阶段的机器翻译系统在翻译一些比较规范的短句子时，效果通常还可以接受，但对于现实生活中出现的真实句子，特别是长句子，翻译效果还是远远不能令人满意。

20 世纪 80 年代末到 90 年代，出现了两种基于语料库的机器翻译（corpus-based machine translation）方法：基于实例的机器翻译（example-based machine translation，EBMT）方法和**统计机器翻译**（statistical machine translation，SMT）方法。到 1999 年以后，统计机器翻译方法取得了很大进展。由于统计机器翻译研究采用大规模的共享数据作为共同的研究基础，并通过定期公开评测进行交流和研讨，还有很多研究者将他们的研究成果以开放源代码形式公开出来，这使得统计机器翻译研究发展非常迅速，成为研究的主流和热点。很多新的理论、方法和模型被提出来并被验证，机器翻译研究重新呈现出一种繁荣的局面。一些统计机器翻译方法也开始融入基

于规则的**知识表示**形式，基于规则的方法和统计方法出现了融合的趋势。统计机器翻译方法的迅速发展，使得机器翻译的水平有了较大的提高。**互联网**的广泛应用，一方面使得机器翻译的需求更为迫切，同时也使得大规模平行语料库的获得变得更加容易。由于统计机器翻译系统可以用语料库自动训练得到，大大缩短了新语种机器翻译系统的开发周期。Google 公司采用统计机器翻译技术，已经可以通过互联网免费提供 50 多种语言的翻译服务。对于某些语序和词义比较相似的语言，通用的机器翻译系统已经达到很高的准确率。但对于一些语序和词义差别较大的语言（如英语和汉语），机器翻译系统准确率仍然不能令人满意。对于平行语料库缺乏的语言对，统计机器翻译也很难发挥作用。另外，领域的适应性是统计机器翻译方法比较难处理的问题，如果训练语料库与应用场合领域差别较大，统计机器翻译也很难达到好的效果。

方　法

可以从不同角度对机器翻译的方法进行分类。

根据机器翻译中源语言到目标语言转换操作所发生的层面，机器翻译方法可以分为直接机器翻译方法、基于转换（transfer-based）的方法和基于中间语言（interlingua-based）的方法。

直接机器翻译方法是最简单的机器翻译方法，是以词典为依托，直接编写程序进行翻译的方法。在这种方法中，除了词典以外的所有翻译知识都直接编写到程序代码中。

在早期的机器翻译系统研制过程中，研究者们意识到，要翻译好一个句子，必须对句子进行一定程度的“理解”，于是提出了一种基于转换的机器翻译方法。在这种方法中，机器翻译被分解成三个步骤：第一步是“分析”，就是将线性序列形式的源语言句子转变成某种形式的深层结构，第二步是“转换”，就是将源语言的深层结构转变成目标语言的深层结构，第三步是“生成”，就是将目标语言的深层结构转变成线性序列形式的目标语言句子。根据转换时所使用的语言深层结构表示层面的不同，基于转换的机器翻译方法又分为基于词（word-based）的方法（这种方法有时也被认为属于直接机器翻译方法），基于短语（phrase-based）的方法、基于句法（syntax-based）的方法和基于语义（semantic-based）的方法。现在的机器翻译系统大部分都采用了基于转换的思想（见图 1）。

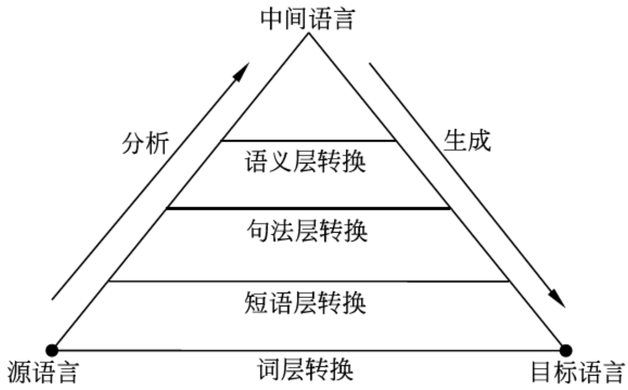


图 1　根据转换层面划分的机器翻译方法

在多语言机器翻译中，为了降低系统的复杂度，可以定义一种独立于所有语言的知识表示形式，称为**中间语言**（interlingua）。这样对于任何两种语言的翻译，只有分析（源语言到中间语言）和生成（中间语言到目标语言）两个过程，省去了转换过程。这种机器翻译方法称为基于中间语言的翻译方法。由于中间语言通常是一种复杂的知识表示形式，需要引入大量世界知识，所以基于中间语言的翻译方法有时也被称为基于知识的翻译方法。在多语言机器翻译的工程实践中，基于中间语言的翻译方法并不成功，原因在于中间语言作为人工设计的知识表示形式，要承载多种自然语言的千变万化的各种语言特性，其工程上的复杂度是难以承受的。不过在一些语义可以清晰定义的窄领域中，这种方法也是可行的，比如多语言的数词翻译，可以采用阿拉伯数字作为中间语言。

在多语言机器翻译中，还有一种常见的做法，即采用一种自然语言（如英语）或者人工语言（如世界语）作为中介，这种作为中介的语言称为**中介语**（pivot language）或者**桥接语**（bridge language）。这种方法也称为基于中介语的机器翻译方法或基于桥接语的机器翻译方法。在这种方法中，一个完整的机器翻译过程被分解成源语言到中介语（桥接语）的翻译和中介语（桥接语）到目标语言的翻译两个过程。

根据机器翻译中所使用的知识表示形式，机器翻译方法又可以分成直接机器翻译方法、基于规则的机器翻译方法和基于语料库的机器翻译方法。

基于规则的机器翻译方法，与直接翻译方法相比，区别在于采用规则库来描述机器翻译的知识。规则库由语言工程师编写，而计算机程序只需要根据规则库的指示进行相应的操作就可以进行翻译了。

规则方法的优点是程序与规则的有效分离,软件设计者不需要了解太多翻译知识,而语言工程师不需要关注算法是如何执行的。这种方法的缺点是翻译规则完全由人工编写,系统开发周期长,而且人工编写的规则库对实际语言现象的覆盖率低。

基于语料库的机器翻译方法,又可以细分为基于实例的机器翻译方法和**统计机器翻译**方法。其主要思想是,不需要人工编写规则库,而是从人工翻译好的双语平行语料库中自动获得翻译知识。

语料库方法的优点是无须人工编写翻译规则,大大减少了机器翻译系统开发的工作量和开发周期,同时翻译知识的覆盖率也大大提高。缺点是翻译系统的性能受制于训练语料库的规模和领域,训练语料库规模不大时数据稀疏问题严重,要翻译的数据与训练语料库领域不一致时性能下降严重。

基于实例的机器翻译方法主要是采用相似替换的想法。首先需要对训练语料库进行加工,得到一个实例库,用于存储大量对齐的句子和句子片段,并提供快速查找的索引。对于要翻译的句子,在实例库中寻找相似的句子或者句子片段组合,利用实例库中对应的译文或者译文片段,构造出被翻译句子的译文。在基于实例的机器翻译中,翻译知识表现为对齐的实例库。

统计机器翻译方法的基本思想是定义一个翻译模型,也就是给定源语言句子后任何目标语言句子的翻译概率。模型的参数可以从双语平行语料库自动学习得到,这个过程称为训练。而有了模型和参数以后,翻译的过程就变成在所有可能的译文句子所构成的空间中搜索翻译概率最大的句子的过程,这个过程称为解码。在统计机器翻译方法中,翻译知识表现为统计模型的参数。

展　望

机器翻译研究的一个趋势是统计方法与语言学知识更加深度的融合。一方面需要采用更加强大的统计模型和机器学习方法,另一方面需要在统计模型中融入越来越多的语言学知识。二者缺一不可。没有更全面深入的语言知识,统计模型很难把握语言本身的深层次规律;而没有更强大的统计模型,就无法解决语言知识之间的矛盾冲突等问题,语言知识不可能得到有效充分的运用。更复杂的语言学知识需要更强大的统计模型才能驾驭。从语言学知识角度看,目前的统计模型已经可以较好地融入句法知识,但更复杂的语义知识和篇章知识在统计机器翻译中应用还不太成功。从统计模型角度看,目前的统计模型还是建立在符号序列和树形结构上,对于在更复杂的结构上如何有效建模还有待进一步研究。

复杂形态语言的机器翻译也是机器翻译研究下一步需要解决的一个问题。对于一些形态复杂的屈折语和黏着语来说,一个词在理论上可能的词形变化有数百种乃至数千种,在这种情况下采用目前的统计模型会导致非常严重的数据稀疏问题。目前统计机器翻译对这一问题还没有很好的解决办法。

语言资源缺乏也是机器翻译需要解决的一个重要问题。目前的主流的机器翻译方法需要使用大量的语言资源(尤其是双语平行语料库)来训练统计模型。而对于世界上大部分语言对来说,大规模的双语平行语料库是很难获得的。如何在双语语料库很少的情况下,尽可能低成本地快速构造一个机器翻译系统,也是机器翻译研究者非常关注的一个课题。

机器翻译的自动评价方法也将是今后机器翻译研究的一个重要课题。目前的机器翻译自动评价在机器翻译水平很低的时候与人工评价还可以保持较好的一致性,但当机器翻译水平逐步提高,乃至接近人工翻译的水平时,机器翻译自动评价的准确性会逐步降低。研究如何提高机器翻译自动评价的准确性,对于引导机器翻译研究向更高水平发展具有重要意义。

参考文献

1. Koehn P. Statistical machine translation. Cambridge University Press, 2010

2. 刘群. 汉英机器翻译若干关键技术研究. 北京:清华大学出版社,2008

3. 宗成庆. 统计自然语言处理. 2 版. 北京:清华大学出版社,2013

(刘群)

jiqi fanyi xitong pingjia

机器翻译系统评价(evaluation of machine translation system) 对**机器翻译**系统的性能的评价。机器翻译系统的性能包括译文质量、翻译速度、人机界面友好程度、健壮性、可维护性等诸多方面。其中,译文质量的评价是机器翻译系统评价中最重要也是最有特色的内容,应用也最为广泛。对于译文质量的评价可以分为人工评价和自动评价两种。人工评价是由人工根据一定的评价指标对翻译系统输出的译文进行打分。自动评价是由评价系统根据一定的数学模型对机器翻译系统输出的译文自

动计算得分。编制一个适当的测试集是译文质量评价,甚至也是其他性能指标(如翻译速度)评价的基础。测试集既独立于具体的机器翻译系统,又独立于具体的评价方法。测试集除了包含源语言例句外,也需要附上正确的至少是可供参考的译文。

早期的机器翻译译文质量评价主要采用人工评价方法。人工评价常用的评价指标是忠实度(adequacy)和流利度(fluency)。忠实度用于评估译文是否忠实地表达了原文的内容,从“完全没有译出来”到“译文准确完整地表达了原文信息”可以划分成多个等级。流利度用于评估译文本身是否流畅和地道,从“完全不可理解”到“译文流畅而且地道”可以划分成多个等级。最后的得分是所有打分的算术平均值。人工方法得到的译文评价结果比较准确,主要问题是成本高,周期长,评价结果也会随着评价人的变化和时间的推移而不同,这使得评价结果不可重复,缺乏客观性。

随着统计机器翻译技术的迅速发展,自动评价逐渐成为译文质量评价的主流方法。自动评价的关键是如何定量地计算翻译系统的译文与一个或多个人工翻译的参考译文之间的相似程度。目前使用最为广泛的自动评价指标是 BLEU(bilingual evaluation understudy)。BLEU 通过比较机器翻译系统译文的 *n* 元语法与参考译文的 *n* 元语法相匹配的程度来综合计算评价得分,匹配的 *n* 元语法的个数越多,BLEU 得分越高。这里 *n* 元语法可以理解为 *n* 个连续的同现词。类似的评价指标还有 NIST,GMT,METEOR 等。相似度的另外一种常见计算方法是编辑距离,即将机器翻译系统译文修改成参考译文需要进行的插入、删除、替换或相邻词交换而进行操作的最少次数,相关的评测指标有 mWER,mPER 等。除了基于相似度的方法外,还有一类引入语言学知识的自动评价方法。比如基于测试点的机器翻译系统自动评价方法,通过分析机器翻译以及两种语言自身的特点,将一个完整译文的评价问题分解成一个个单独的测试点的评价,每个测试点本身的评价只要通过简单的字符串匹配即可实现。这类方法可以从评价结果中分析出机器翻译系统在哪些语言问题上处理得不够好,从而有针对性地对系统进行改进。自动评价方法的引入有效减少了人工评测的主观性,有助于研究人员快速测试机器翻译系统的改进效果,从而缩短机器翻译统的研发周期,对机器翻译技术和系统的研究与发展起到了很大的推动作用。

除了对译文质量进行评价之外,还可以从其他方面对机器翻译系统的性能进行评价,如翻译速度、人机界面的友好程度、对于错误输入的处理能力,词汇和语法是否便于扩充等。不同的应用目标、不同的用户所关心的评价角度与内容也会有所不同。综合使用各种标准进行多方位的评价,才有可能得到比较全面、客观的评价结果。

参考文献

1. 宗成庆. 统计自然语言处理. 2 版. 北京:清华大学出版社,2013

2. 冯志伟. 自然语言处理的形式模型. 合肥:中国科学技术大学出版社,2010　(吕雅娟　刘群)

jiqiren chuanganqi

机器人传感器(robot sensor)　用于测量机器人自身及其周边环境状态信息的装置。主要分为两类:内部状态传感器和外部状态传感器。

内部状态传感器主要用来测量机器人的内部状态,如机器人关节位置和速度变量。最典型的如安装在电机轴上的光码盘,它用来直接测量机器人的关节角位移。外部状态传感器主要用来测量机器人与外界环境之间相互关系的变量信息,如力觉、触觉、视觉、声觉、接近觉等。机器人传感器研究目前主要关注外部状态传感器。而且,由于机器人视觉已发展成为相对独立的研究领域,若不特别说明,机器人传感器主要是指非视觉传感器,即力觉、触觉、接近觉等。

力和力矩传感器　用于机械手的力传感器。安装位置可以在机械手的关节处实现关节力矩测量,或者在被操作物的台架上,或者在机械手的末端处实现末端作用力测量。

(1)关节力/力矩传感器　在上述三种传感器形式中,关节力/力矩传感器是应用得最早的一种。早期主要用在主从机械手中,测量操作人员操纵主手时力的作用。对于电动机械手,关节力矩可以直接通过测量驱动电机的电流来获得。由于关节力矩的测量分布于整个机械手,它不仅可以检测手部受到的力,同时也能测出机械手其他部分所受的力。通过换算,可以获得机械手末端的作用力,但精度较差。

(2)台架力/力矩传感器　若将力测量装置安放在机械手所要操作物体的台架上,机械手的末端力也可以通过这种方法直接测量出来。它主要适用于利用机械手进行研磨、切削等工作。该方法的主要缺点是灵活性较差,它只适用于某些特定工作的

情况。

(3) 腕力传感器　力传感器安装在机械手的手腕处,故称腕力传感器。它直接测量机械手末端的作用力,精度较高,是目前应用得较多的一种。腕力传感器通常要求体积小、重量轻和灵敏度高。腕力传感器的典型结构是在一个十字花梁的四根挠曲棒粘贴八对半导体应变片,棒端的每一面均贴有一片。棒端对面的两边应变片连到差动电位器电路,其输出电压与垂直应变片平面方向的力成正比。差动连接可补偿温度变化的影响。八对应变片所产生的八组输出读数,根据力和力矩的平衡条件,解算出沿坐标轴方向的三个力分量和三个力矩分量。

触觉传感器　触觉传感器主要用来测量机械手与所操作物体接触的有关信息,可用于物体的定位和识别,也可用来控制对物体施加的力。触觉传感器可分为两大类:开关信号触觉传感器和连续信号触觉传感器。开关信号触觉传感器是由诸如微动开关那样的接触装置所组成,形成信号通路的闭合和断开。安装于手指的内表面时,不仅可感知手指中间是否有物体存在,而且经过多次触摸,可以帮助机械手确定抓取位置。连续信号接触传感器采用柔顺装置,它的输出正比于局部作用力的大小。常用的方法有:弹性杆测量,主要用来测量一个点的接触力;触觉传感阵列,由若干个感知单元组成传感阵列,可提供较大范围的触觉信息;人工皮肤,通过感知接触变形来获得局部的力信号。

上述传感器测量的是法线方向的力信号,如果测量的是切向力,则它可用作为滑觉传感器。

接近觉传感器　接近觉传感器主要用来感知在一定近距范围内是否有物体存在,用于机器人抓取物体和避碰,避免接近速度过快造成冲击。有以下几种实现方法。

(1) 感应传感器　其工作原理是当附近存在金属物体时,引起传感器电感的变化,从而可测知物体的存在。

(2) 霍尔效应传感器　它利用霍尔效应原理来感知附近物体的存在,但只适用于铁磁材料物体,其应用受到一定限制。

(3) 电容传感器　当附近存在物体时,将引起电容的变化,从而感知物体的存在。

(4) 超声传感器　通过检测反射的声波来感知周围物体的存在,不受物体材料性质的限制。

(5) 光学传感器　通过检测反射光来感知周围物体的存在。

距离传感器　它用来测量从参考点(通常是传感器本身)到物体间的距离,主要用于移动机器人的导航和避碰。有以下几种实现方法。

(1) 三角测量法　光源发射光束,改变发射角直至接收器接收到在附近物体反射的光束,由三角关系算出到物体的距离。

(2) 结构光方法　通过发射出一定的结构光模式(常用窄光面激光)到附近的物体,物体上显示出一条光带,摄取该光带信息并送至计算机分析即可获得距离信息。

(3) 激光测距　通过测量发射与接收激光之间的飞行时间间隔来计算被测物体的距离。

(4) 超声测距　通过测量发射与接收超声波之间的飞行时间间隔来获取距离信息。

机器人传感器是传感器技术在机器人中的具体应用,是机器人感知外界信息的输入接口,是实现机器人智能化的必要工具和手段。目前,机器人传感器正朝着高精度、小型化、集成化和智能化的方向发展。

参考文献

1. Fu K S, Conzalez R C, Lee C S G. Robotics: Control, sensing, vision and intelligence. McGraw-Hill Book Company, 1987

2. 高国富,谢少荣,罗均. 机器人传感器及其应用. 北京:化学工业出版社,2005

（孙增圻　杨唐文）

jiqiren kongzhi

机器人控制(robot control)　以研究机器人为对象的控制原理和技术。

机器人的动力学模型具有强耦合和非线性的特点,是一个难以控制的复杂对象。机器人控制的主要问题在于找到性能优良、易于实现的控制方法。目前,机器人主要采用独立关节的PID伺服控制,它具有稳定性好、控制简单的特点。但当机器人的运动速度较高时,该方法的缺点便明显表露出来。科研人员已从不同的角度对机器人的控制问题进行了比较深入的研究,且提出了各种不同的方法,比较典型的有:分解运动速度控制、分解运动加速度控制、计算力矩控制、滑模变结构控制、非线性控制、自适应控制等。

机器人的运动控制通常划分为两个阶段。第一阶段是从起始位置到达目标点附近沿期望轨迹的粗略运动,主要是位置的控制;第二阶段是精细的运动

控制，此时，机器人末端执行器对物体进行操作，完成要求的作业任务，同时需要增加外部力觉反馈的柔顺运动控制。它不同于纯粹的位置控制，是机器人控制中的另外一个重要内容。机器人柔顺控制主要采用阻抗控制和混合位置/力的控制方法。

由于对机器人的精确建模比较困难，尤其是考虑摩擦等实际的非线性因素时更加如此。因此，人们对于不太依赖模型的机器人智能控制方法日益感兴趣。其中，较为突出的是机器人模糊控制和神经元网络控制，它们越来越受到人们的关注和重视，并已显示出广泛的应用前景。

机器人控制方法的研究初始于20世纪70年代。80年代，对各种新的控制方法研究达到高潮。这些方法多数处于研究阶段，真正实际应用的并不是很多。其主要原因是它对模型的要求较高，或者算法过于复杂，难以实时实现。目前，机器人控制更多集中在基于多传感器信息融合的智能控制方法的研究。

机器人控制的工程实现与一般控制相类似，主要采用计算机控制的结构形式，以实现更为复杂和更高要求的控制。

参考文献

1. 孙增圻，严隽薇，钱宗华. 机器人智能控制. 太原：山西教育出版社，1994

2. 谭民，徐德，侯增广，等. 先进机器人控制. 北京：高等教育出版社，2007

（孙增圻　杨唐文）

jiqiren yundong guihua

机器人运动规划（robot motion planning）

指在满足物理约束条件下，按照一定的评价标准，设计机器人从初始状态到达目标状态的动作序列的过程和方法。

对于**智能机器人**领域中最常见的研究对象——机械手和**移动机器人**来说，所要满足的物理约束主要是避免与环境中的其他物体发生碰撞。在这种意义下，运动规划问题可描述为：假设 A 为在欧氏空间 W（工作空间）中一个刚性运动物体——机器人；$B_1, B_2, \cdots, B_q$ 为分布于 W 中的固定或运动物体，称为障碍；如果 $A, B_1, B_2, \cdots, B_q$ 的几何参数以及 B_i 在 W 中的位置是精确已知的，且 A 的运动不受任何动力学的约束，那么，给定 A 在 W 中的初始状态和目标状态（位置和方位），要求产生一条路径 P，它确定一个使 A 从初始状态开始，于目标状态结束，且不与 B_i 发生碰撞的运动状态的连续序列。而且，如果这种路径存在的话，给出在某种意义下的最优解。

机器人运动规划水平是体现其智能的重要标志，也一直是智能机器人研究领域的热点和难点。机器人运动规划的研究涉及**人工智能**、计算机科学、数学、机械工程等多学科领域。由于决定机器人运动规律的逆运动学是一个多维多值的非线性函数，而且机器人和障碍物通常是具有复杂几何形状及多自由度的三维物体，因而对运动规划的求解提出了较高的要求，也增加了算法的复杂性。这方面的典型结论包括：三维多面体的环境中任何 L^p 度量下寻找最短路的问题是 NP-HARD；二维环境中具有运动障碍（若运动物体的速度有限）的运动规划是 NP-HARD。

机器人运动规划的早期研究可以追溯到20世纪60年代末期。Howden 于1968年发表题为“The sofa problem”的文章，“sofa”指的是一个二维物体 A，环境为一个复杂的二维结构 M，所要解决的问题是物体 A 能否在 M 内从一点 P_1 运动到另一点 P_k。虽然 Howden 对运动规划问题的讨论作了诸多简化，但其中思想和算法却蕴含了后来对规划问题所发展出的各种技术的基本思想，特别是关于运动规划问题的一般求解结构：①划分空间；②建立包含划分信息的网络；③在网络上搜索路径。

20世纪80年代出现了大量具有重要理论和实际意义的研究工作。Lozana-Perez 系统地提出了利用构形空间来求解机器人无碰撞路径的方法。其基本思想是将机器人描述为一个点，而把环境障碍物看成为构形空间中的多面体，那么机器人的路径规划转换成点在构形空间中寻找最优路径的问题。典型算法包括势场法、可视图法、Freeway 算法、Subdivision 算法和收缩法等。势场法曾在机器人的路径规划中广泛应用。它将机器人的运动视为工作空间的虚拟力场中合力作用下的加速度运动。力场的虚拟力来自障碍物体对机器人产生的斥力和目标位置产生的引力。

近十几年，基于栅格图的搜索算法在机器人运动规划中得到广泛应用。栅格图实质上是运动区域的立体地图，是被横线、竖线划分成的一系列单元。每个单元根据障碍信息赋予不同的数值，即穿越该单元所付出的成本。栅格图建立以后，机器人的运动规划问题则可以看成是在立体地图上寻找总的穿越成本最低的运动路径过程。建立栅格地图时，栅格单元的空间大小、数量以及成本值需认真加以考虑。栅格单元的定义对搜索算法的影响很大，包括搜索时间和占用的计算资源。

机器人运动规划的主要研究方向包括：针对高维空间运动规划实时性差的缺陷，研究快速在线的运动规划方法；针对运动规划中鲁棒性较差等局限性，在感知空间中研究运动规划问题，如基于传感器的感知行为和反射式规划的研究，**机器学习**和传感器自组织等问题的研究；针对环境模型和机器人运动控制的不确定性，研究分析和处理环境模型误差和机器人的控制误差等。近几年，基于随机采样的机器人运动规划方法受到关注，它有效地解决了高维空间和存在复杂约束的运动规划问题。

参考文献

1. Howden W E. The sofa problem. Computer Journal,1968,11：299-301

2. Lozano-Perez T. Spatial planning：A configuration space approach. IEEE Trans on Computers,1983,32：108-120

3. Chien R T, Zhang L, Zhang B. Planning collision-free paths for robotic arm among obstacles. IEEE Trans on pAMi,1984,6：91-96

4. LaValle S. Planning algorithms. Cambridge University Press,2006　（王宏　杨唐文）

jiqishu

机器数（machine number）　计算机中按一定规则表示的数据。它涉及记数制、定点数和浮点数的表示，二进制数的原码、补码、反码的表示等（参见**数制**）。

除此以外，在计算机中可能还包括应用或控制使用的数据，例如数字化的图像、语音信息以及控制用的逻辑数等。

对于定点数，一个 n 位的二进制，可以有 2^n 个不同的状态，也可以表示 2^n 个不同值的数。但是因为原码的零有正零（00…00）和负零（10…00）两种形式，反码的零也有正零（00…00）和负零（11…11）两种形式，所以它们能代表的数的个数为 2^n-1 个，而补码的零仅有一种表示形式，即正零（00…00），所以能代表数的个数为 2^n 个。

浮点数在计算机中通常以规格化的形式存储，当运算结果小于机器能表示的最小规格化数时，则以机器零表示，即阶码及尾数均设置为全 0。

（王爱英）

jiqi xuexi

机器学习（machine learning）　计算机模拟或实现人类的学习行为，以获取新的知识、技能或者重新组织已有的知识结构，使之不断改善自身性能的理论、模型和方法。它是**人工智能**领域的一个重要分支。

学习能力是人类智能行为的一个非常重要的特征，但至今对学习的机理尚不清楚。人们曾对机器学习给出各种定义。H. A. Simon 认为，学习是系统所作的适应性变化，使得系统在下一次完成同样或类似的任务时更为有效。R. S. Michalski 认为，学习是构造或修改对于所经历事物的表示。从事专家系统研制的人们则认为学习是知识的获取。这些观点各有侧重，第一种观点强调学习的外部行为效果。第二种则强调学习的内部过程，而第三种主要是从知识工程的实用性角度出发的。

机器学习在人工智能的研究中具有十分重要的地位。一个不具有学习能力的智能系统难以称得上是一个真正的智能系统，但是以往的智能系统都普遍缺少学习的能力。例如，它们遇到错误时不能自我校正；不会通过经验改善自身的性能；不会自动获取和发现所需要的知识；它们的推理仅限于演绎而缺少归纳，因此至多只能够证明已存在的事实、定理，而不能发现新的定理、定律和规则等。随着人工智能的深入发展，这些局限性表现得愈加突出。正是在这种情形下，机器学习逐渐成为人工智能研究的核心之一。它的应用已遍及人工智能的各个分支，如**专家系统**、**自动推理**、**自然语言理解**、**模式识别**、**计算机视觉**、**智能机器人**等领域。其中尤其典型的是专家系统中的**知识获取**瓶颈问题，人们一直在努力试图采用机器学习的方法加以克服。

机器学习的研究是根据生理学、认知科学等对人类学习机理的了解，建立人类学习过程的计算模型或认知模型；发展各种学习理论和学习方法，研究通用的学习算法并进行理论上的分析；建立面向任务的具有特定应用的学习系统。这些研究目标相互影响相互促进。

发展简史

自 1956 年人工智能奠基以来，人们非常重视机器学习的研究，至今已经历了四个发展阶段。

第一阶段始于 20 世纪 50 年代中期，这一阶段的工作不少程度上受启发于神经生理学、生物学等研究，主要是研究不需要什么初始知识的通用学习系统，特别是神经网络系统。这一阶段的一个重要特点是数值表示和参数调整，不像当时的人工智能领域重心在于符号表示和启发式方法；而更偏向于

模式识别。这一时期的代表性工作有**感知机**、生物进化过程模拟,以及 A. L. Samuel 的很有名的、曾击败过州级冠军的计算机跳棋学习程序。

第二阶段始于 20 世纪 60 年代初期,这一阶段的不少工作受到心理学和人类学习的启示,主要是概念学习和语言获取,有人称其为符号概念获取阶段。这一时期的主要特点是使用符号表示而不是数值表示。当时符号表示已成为人工智能的主要方法。另外,采用数值表示的神经网络由于 M. Minsky 等于 1969 年发表的著作从理论上分析了感知机的能力和限制,使得这方面的研究陷入低谷。这一时期的代表性工作有概念学习系统 CLS,积木世界结构学习系统。另外,在学习计算理论方面,建立了极限辨识理论。

第三阶段始于 20 世纪 70 年代中后期,由于专家系统的成功和**知识工程**的形成,这一阶段的工作对知识的重要性尤其关注。一方面,领域知识大量引入学习程序之中;另一方面,知识的自动获取成为机器学习的应用目标。这一时期,机器学习逐渐走向兴盛,各种学习策略、学习方法相继出现,除了作为主流的归纳学习外,还出现了**类比学习**、解释学习、观察和发现学习等等。这一时期有影响的工作有学习质诺仪预测规则系统 Meta-DENDRAL,利用 AQll 方法学习大豆疾病诊断规则系统,利用 ID3 方法学习象棋残局规则,数学概念发现系统 AM,符号积分系统 LEX,以及一系列物理定理重新发现系统 趴 ON。在学习计算理论上,L. G. Valiant 提出了概率近似正确 PAC 学习模型,这一成果推动了学习计算理论的发展。于 2010 年 L. G. Valiant 获得了 ACM 图灵奖。

第四阶段始于 20 世纪 80 年代中后期,主要源于**人工神经网络**的重新兴起。由于使用隐单元的多层神经网络及反传算法的提出,克服了早期线性感知机的局限性,从而使得非符号的神经网络的研究得以与符号学习并行发展。同时,机器学习在**符号学习**的各个方面也更加深入和广泛地展开,并形成了较为稳定的几种学习风范,如归纳学习、分析学习(特别是解释学习和**类比学习**)、遗传学习等。这一时期有影响的工作有多层神经网络反向传播学习算法,基于解释的学习,一系列**决策树**归纳学习方法,J. H. Holland 的遗传学习和分类器系统,A. Newell 等的 SOAR 学习系统,以及 PRODIGY 学习系统等。近期,由于复杂世界的实际应用的需要,出现了结合各种学习方法的**集成学习**系统、多策略学习技术,特别是关于连接学习与符号学习的结合。1989 年 8 月举行的第 11 届国际联合人工智能学术会议上首次出现了数据集知识发现(Knowledge Discovery from Dataset,KDD),把机器学习与**知识库**研究结合起来,极大地推动了机器学习的理论研究,展示机器学习极大的应用价值。

学习系统模型

机器学习经过三十多年的发展,到现在已形成了很多学习方法,例如机械学习、传授学习、实例学习、发现学习、解释学习、**类比学习**、**强化学习**、事例学习、遗传学习、连接学习等。这些学习方法可以用一个学习模型来描述(参见图 1)。

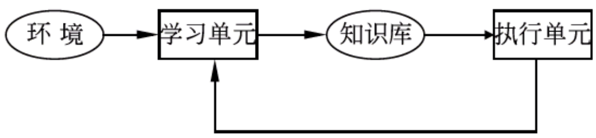


图 1　一个简单学习系统模型

在图 1 中,椭圆圈表示信息体(如观察的数据,以及事实、规则等知识),方框表示过程。箭头指示数据在学习系统中的流向。环境为学习单元提供外界信息源(如经验实例)。学习单元利用该信息对知识库作出改进(增加新知识或重新组织已有知识)。学习单元算法从训练样本集产生建立模型,有时被称为学习器(learner)。执行单元利用知识库中的知识执行任务,任务执行后的信息又反馈给学习单元作为进一步学习的输入。

学习单元的输入有两种:一是外界环境,另一是执行任务后的反馈信息。不同的学习系统有不同的经验实例表示。最简单的一种是二元特征表示,仅仅描述对象某些属性的存在与否,例如病人有或没有某个特定症状。下文要讲的连接学习和遗传学习方法一般使用这种二元特征的输入。另一种是用属性值表示,每个属性有一组相互排斥的值,如颜色属性的值可为红色、蓝色和黄色等。二元特征可看作是此类的特例。这种属性值表示典型地用在归纳学习方法中。还有一种更复杂的表示是关系或结构表示,描述两个或多个对象间的关系,如对象 A 位于对象 B 的上方。这种关系或结构信息一般是以谓词逻辑、语义网络等形式表示的。同前两种表示相比,这种表示具有更强的表示能力,但同时也为作为学习中重要部分的匹配过程带来了相当的复杂度,以致影响了它们的使用。分析学习主要处理这种关系表示型的数据结构。

知识库用来存储知识，包括系统原有的领域知识（这种知识是长期的、相对稳定不变的），以及通过学习而获得的各种新知识（这种知识是短期的、相对不稳定、变化的）。选择何种**知识表示**对学习系统的设计起着非常重要的作用，如是存储具体的单个的经验实例还是仅存储从这些实例得到的抽象推广。机器学习的大部分方法是只存放总结了以前经验的、普遍性的知识，而分析学习中的类比学习及基于范例的学习则是混合了两种方式。如果知识是后一种表示（以抽象概括的形式存储），则存在两种区别，即是以逻辑的、离散的形式表示，还是以数值的、连续的形式表示信息。归纳学习和分析学习使用的是离散的、逻辑的表示方法，连接学习则使用数值的、连续的表示，遗传学习则是两种表示的结合。

执行单元既是使得学习系统具有实际用途，又是评价学习算法性能好坏的关键部分。机器学习研究的很大一部分工作是集中在这样两个领域：分类和问题求解。一个智能主体常常需要将其经验或经历的事例进行分类标记。例如，在面临病人表现的某些特定症状时，医生需要作出某一疾病的诊断。机器学习至今所作的很大一部分是学习分类。该任务可描述为，给定一个经验实例的某种描述，以及一些已知的类，任务在于试图将这些经验实例赋予某个类别。专家系统所作的最常见的也是最有用的工作便是同这种分类或诊断有关。另外，机器学习也常要能解决新问题或进行规划，例如，任务是设计一个从当前所在地到达目的地的路径。一般是给定一些所要达到的状态或目标，任务在于试图找出一个动作序列，使得其主体能从当前状态达到目标状态。机器人操作是最明显的问题求解策略的应用。这个领域涉及技能的获得，用以改进问题求解的速度和规划推理的能力。

学习单元是学习系统内实现学习算法功能的核心。一般涉及这样两个方面：①一种是处理经验事例的方式，有渐进式和非渐进式。渐进式是指每次仅处理一个事例，能不断地处理新遇到的事例。因而，这种方式能处理很大的（理论上是无穷大的）数据集。非渐进式则指仅用一次时间处理有相当大小的一个事例集，这种方式的长处是能根据大量数据的统计特点获得很多决策、推断信息。②另一种是获取新知识过程中所用的推理方法，主要有归纳和演绎方法，也有反绎方法。归纳方法主要基于观察数据来形成一般化的知识，它的特点在于产生的知识是先前知识库中所没有的（或不能蕴涵的）。通过归纳产生的模型适用于新样本的能力被称为泛化（generalization）能力，提高泛化能力是机器学习的根本问题。归纳方法主要用于归纳学习、连接学习和遗传学习等学习风范。演绎方法则依赖于知识库中已有的知识来形成新知识，如基于解释的学习是利用先前的知识来解释新的事例，然后简化该解释并存放于知识库中。这种方法主要用于分析学习中；另外，归纳和演绎方法的结合，比如类比学习能得到更好的学习效果。

当考虑执行单元的反馈信息时，还存在两种相对的学习方式，即**监督学习**和**非监督学习**。前者是指有导师立即给予学习者（学习单元）关于其学习行为的反馈信息，后者则是学习者得自己为自己作出判定，或仅能得到一点很粗略的指导。在分类任务领域里，监督学习一般称为实例学习。在分类任务中，非监督学习的学习者必须自己决定自己的类以及类的数目。这种学习称为概念聚类。在问题求解领域中，监督学习指的是导师能在求解的每一步搜索过程中向学习者建议正确的操作或子目标。在这方面，研究得更多的是非监督学习，学习者得靠自己区分哪种动作是自己所希望的，这种方法称作信任赋值。

目前，机器学习风范主要有归纳学习、类比学习、强化学习、分析学习、遗传学习、连接学习等。

归纳学习

归纳学习从具体实例出发，通过**归纳推理**，得到新的概念或知识。归纳学习的基本操作是泛化和特化。泛化是使规则能匹配应用于更多的情形或实例。特化操作则相反，减少规则适用的范围或事例。归纳学习是目前研究得最广泛的一种**符号学习**方法，包括实例学习、概念聚类、发现学习等。实例学习的任务是，给定关于某个概念（或多个概念）一系列的已知的正例和反例，要求从中归纳出一个（或多个）一般的概念描述，该描述能使得这些已知的正例可从中再次推导出来而同时没有任何反例可从中推导出来。概念聚类则是由程序根据实例间的某种类似度关系自动形成有用的概念描述（参见归纳学习）。发现学习主要是从实验数据、观察实例或数据库中获得知识。

类比学习

类比学习是基于类比推理的一种学习方法。对于两个对象，如果它们之间有某些相似之处，那么就推知这两个对象间还有其他相似的特征。类比学习系统就是通过在几个对象之间检测相似性，根据源

对象所具有的事实和知识,推论出相似对象所应具有的事实和知识。基于案例推理、迁移学习等都是基于类比学习的原理。

强化学习

人类通常从与外界环境的交互中学习。强化学习是指从环境状态到行为映射的学习,以使系统行为从环境中获得的累积奖励值最大。在强化学习中,我们设计算法来把外界环境转化为最大化奖励量的方式的动作。我们并没有直接告诉主体要做什么或者要采取哪个动作,而是主体通过看哪个动作得到了最多的奖励来自己发现。主体的动作的影响不只是立即得到的奖励,而且还影响接下来的动作和最终的奖励。试错搜索和延期强化这两个特性是强化学习中两个最重要的特性。

分析学习

分析学习是利用背景或领域知识,分析很少的典型实例(通常仅一个),然后通过演绎推导,形成新的知识,使得对领域知识的应用更为有效。分析学习方法的目的在于改进系统的效率性能,而同时不牺牲其准确性和通用性,这不同于归纳学习方法。常见的分析学习方法有解释学习等。

遗传学习

遗传学习源于模拟生物繁殖中的遗传变异原则(交换、突变等),以及达尔文的自然选择原则(生态圈中适者生存)。一个概念描述的各种变体或版本对应于一个物种的各个个体,这些概念描述的变体在发生突变和重组后,经过某种目标函数(相应于自然选择准则)的衡量,以决定谁被淘汰谁继续生存下去(参见遗传算法)。

连接学习

连接学习是在**人工神经网络**中,通过样本训练,修改神经元间的连接强度,甚至神经网络本身的结构的一种学习方法。这种学习方法和符号学习并立而行。它与符号学习不同,主要是基于样本数据进行学习(参见人工神经网络)。

计算学习理论

计算学习理论主要研究学习算法的样本复杂性和计算复杂性。这方面的早期成果主要是基于Gold 框架,在形式语言学习的上下文中,Gold 引入收敛的概念,有效地处理了从实例学习的问题。基于 Gold 学习框架,Shapiro 提出了模型推理算法。1984 年 Valiant 提出概率近似正确(Probability Approximation Correct,PAC)的学习框架,它仅要求与目标概念具有高概率的近似,而并不要求目标概念精确的辨识。在概率近似正确学习理论的基础上,Vapnik 发展了统计学习理论,并提出结构学习方法和支持向量机(Support Vector Machine,SVM)。

展　望

机器学习已建立了一定的理论基础和自己的科学研究方法,形成了各种研究风范以及相应的学习算法。但是,目前的机器学习风范均有其局限性。在了解人类学习机制的基础上,正结合具体的应用领域,深入开展学习风范的研究,研究不同学习风范的集成,探讨新的学习方法和算法。

参考文献

1. Mitchell T M. Machine learning. McGraw Hill, 1997

2. 史忠植. 知识发现. 2 版. 北京: 清华大学出版社, 2011

3. 周志华,王珏. 机器学习及其应用 2009. 北京: 清华大学出版社, 2009

(史忠植)

jiqi xuexi zhong de zhengzehua

机器学习中的正则化(regularization in machine Learning) 一种利用先验知识来解决**统计学习**中过拟合(overfitting)问题或数学中病态(ill-posed)问题的有效技术。本质是通过问题先验知识的引导来约束解空间的范围以获取期望的稳定解。该技术已广泛用于**机器学习**、**模式识别**等众多领域。

正则化由 Tikhonov 于 1963 年首先提出,于 1990 年被引入机器学习领域,并于 1995 年获得了与其等价的正则化网络形式。正则化的表示形式多样,基于再生核 Hilbert 空间中的一般性正则化理论已渐趋完善,并形成了基于代数和统计观点的两种基本解释,使众多机器学习算法被纳入同一框架。正则化技术已成为**统计学习**的核心问题之一。

提高学习系统或算法的泛化能力(generalization)是机器学习的中心任务,而"泛化能力很大程度上依赖于数据和知识"。在现实的有限样本下,最小化经验风险是达成学习任务的关键,但其常导致的正是病态问题,使所建系统产生糟糕的泛化性能。借助正则化技术可将先验知识表成正则化项添加至经验风险,避免上述现象发生。

正则化原理:获得稳定解的途径是构造与解的先验知识相符的非负辅助项,实现对解空间 H 的约束,即:

$$\begin{aligned} f_\lambda(x) &= \underset{f_\lambda \in H}{\operatorname{argmin}} R(f) \\ &= \underset{f_\lambda \in H}{\operatorname{argmin}} \left\{ R_{emp}(f) + \sum_{i=1}^{n} \lambda_i R_{reg}^{i}(f) \right\} \end{aligned}$$

其中$f_{\lambda}(x)$是最小化$R(f)$所得最优解，$R(f)$包含两部分：

（1）经验风险部分$R_{emp}(f)$，常定义为$R_{emp}(f)=\frac{1}{l}\sum_{i=1}^{l}L(y_i,f(x_i))$，其中，$L(y,f(x))$为损失函数，度量了$x$的期望输出$y$和实际输出$f(x)$间的差异。

（2）正则化部分$R_{reg}^{i}(f)$（$i=1,2,\cdots,n$），刻画（或反映）问题领域或解的某种先验。常用的先验知识包括：解的光滑性；数据的分布特性和属性的稀疏性等。不同的先验知识可用不同的正则化项来表示。正则化项大致可分为如下类型：解层面的光滑正则化项（如 Tikhonov 正则化$\|f\|_K$），数据层面的结构正则化项（如流形正则化$\|f\|_I$）和属性层面的相关正则化项（如 LASSO 正则化$\|f\|_{lasso}$）等。其中λ_i是个相当关键的因子，称为正则化参数，它折中经验风险和解的复杂性，使解既不过简单又不过复杂。过简单导致差的数据拟合而过复杂导致差的预测或泛化性能。正则化项的加入本质上在于收缩解空间，使所求之解尽量在达到对经验数据充分拟合的同时与先验知识相吻合，如此才能保证有高泛化性能的稳定解或模型。

正则化在理论和应用上均已取得了重要进展，并已衍生出众多经典的机器学习算法，其进一步研究的主要方向包括：

（1）模型选择：关键是正则化参数的选择，其直接影响泛化性能。

（2）正则化项构造：对问题先验知识的合理表示或刻画，对经验风险适度的惩罚与补偿，是产生新颖正则化项的关键。

（3）泛化误差界研究：泛化性能是评价正则化模型优劣的重要指标，具有核心的理论和应用研究价值。

参考文献

1. Tikhonov A N. Solution of incorrectly formulated problems and the regularization method. Dokl. Akad. Nauk SSSR，1963，151(3)：501-504

2. Hastie T，Tibshirani R，Friedman J. The elements of statistical learning：Data mining，inference，and prediction. 2nd ed. Springer-Verlag，2009

3. Simon H. 神经网络与机器学习. 3 版. 申富饶，等译. 北京：机械工业出版社，2009

（陈松灿　花强）

jiqi yuyan

机器语言（machine language）　表示成数码形式的机器基本指令集，或者操作码经过符号化的基本指令集。

机器语言一般由一台机器可以执行的全部指令及其所操作的数据组成，其功能可以通过相应计算机的基本指令集合（也称作指令系统）加以描述。其中，每条指令将指挥计算机执行一个基本操作，包括数据处理操作（如算术运算、逻辑运算、字符处理等），控制操作（如判断、转移、中断、改变机器状态等）和传输操作（如输入、输出、数据移动等）。在计算机中，实施操作的指令和被实施操作的对象均要表示成二进制代码形式。指令由操作码和地址码两部分组成，操作码指明要实施的基本操作；而地址则指明被实施操作的对象在计算机中的存放位置。被实施操作的对象可以是整数、实数、布尔值、字符串等，由于它们最终均要表示成二进制数字序列的形式存放在计算机中，为了区分它们的类型，通常要在操作码中设置“标志”字段来加以标识，这与高级语言中利用类型说明来标识对象的类型是有所不同的。机器语言的主要特点是与特定的机器相关，运行效率较高级语言高，但用户难于使用，烦琐，费时，且易出错。

（曹东启）

jidai chuanshu jishu

基带传输技术（baseband transmission technology）　指不对基带信号进行调制，而直接进行传输的技术。数字信号源经过码型变换后形成数字基带信号，不经过调制，直接送到通信介质上传输，称为数字基带传输。对于模拟信号源，可以直接使用模拟基带传输，也可以将模拟信号源转化为数字信号源，使用数字基带传输系统进行传输。基带传输不需要调制，因此主要用于短距离、不需要复用通信介质的环境。

数字基带传输系统主要由信道信号形成器（包含码型变换器、波形变换器、发送滤波器等）、信道、接收滤波器、同步提取和取样判决器等 5 个功能电路组成，如图 1 所示。

数字信号源的信息首先经过信道信号形成器中的码型变换器，将二进制脉冲序列编码形成基带传输信号，有时还要进行波形变换器和发送滤波器，提高传输信号在信道中的抗码间干扰能力。经过传输信道后，在接收端为了减小噪声的影响，首先使信号进入接收滤波器，然后再经过均衡器，校正由于信道特性（包括接收滤波器在内）不理想而产生的波形失真或码间串扰。最后在取样定时脉冲到来时，进行判决以恢复所传送的信息。

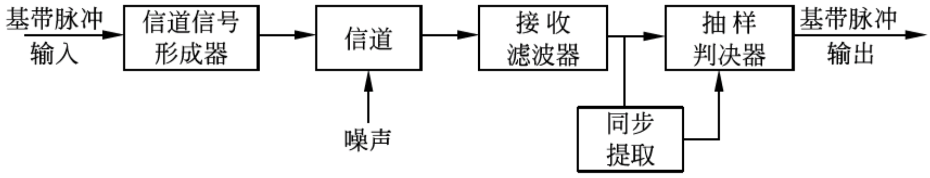


图 1　数字基带传输系统

在实际基带传输系统中，未经编码的原始信号或任意编码的数字信号就不一定适合在信道中传输，例如，含有丰富直流和低频成分的基带信号就不适宜在信道中传输，因为它有可能造成信号严重畸变；再如，一般基带传输系统都是从接收到的基带信号中直接提取位同步信号，如果原始数据出现长时间的连“0”符号，那么未经编码的基带信号可能会长时间出现 0 电位，从而使位同步恢复系统难以及时得到校正。因此基带传输系统的编码（常称传输码，又称为线路码）需要考虑以下原则：

（1）码型中应不含直流分量，低频分量尽量少；

（2）码型中高频分量尽量少，这样既可以节省传输频带，还可以减少串扰；

（3）码型中应包含定时信息；

（4）码型具有一定检错能力；

（5）高的编码效率；

（6）编译码设备应尽量简单。

上述各项原则并不是任何基带传输码型均能完全满足的，往往是依照实际要求满足其中若干项。

数字基带信号的码型种类繁多，仅以矩形脉冲组成的基带信号为例，常见的传输码型有不归零码（NRZ）、归零码（RZ）、传号交替反转码（AMI）、三阶高密度双极性码（HDB3）、双相码（PC）及传号反转码（CMI）。

不归零码又分为单极性不归零码和双极性归零码，不归零码的波形特点是电脉冲之间无间隔，极性单一，如图 2 所示。

归零码又分为单极性归零码和双极性归零码，归零码的特点是信号电压在一个码元终止时刻前总要回到零电平，如图 3 所示。

传号交替反转码的编码规则是将消息码（传号）交替地变换为“＋1”和“－1”，而“0”（空号）保持不变。三阶高密度双极性码是传号交替反转码码的一种改进型，改进目的是为了保持传号交替反转码的优点而克服其缺点，使“0”个数不超过 3 个，如下所示：

消息码　1 0 0 0　0　1　0 0 0　0　1　1
　　　　0 0 0　0　0 0 0　0　1　1

AMI 码　－1 0 0 0　0　＋1　0 0 0　0　－1　＋1
　　　　0 0 0　0　0 0 0　0　－1　＋1

HDB3 码　－1 0 0 0 －V　＋1　0 0 0　＋V　－1
　　　　＋1　－B 0 0　－V　＋B 0 0　＋V
　　　　－1　＋1

传号交替反转码对应的波形是具有正、负、零三种电平的脉冲序列。三阶高密度双极性码的 ±V 脉冲和 ±B 脉冲与 ±1 脉冲波形相同，用 V 或 B 符号表示的目的是为了示意该非“0”码是由原信码的“0”变换而来的。

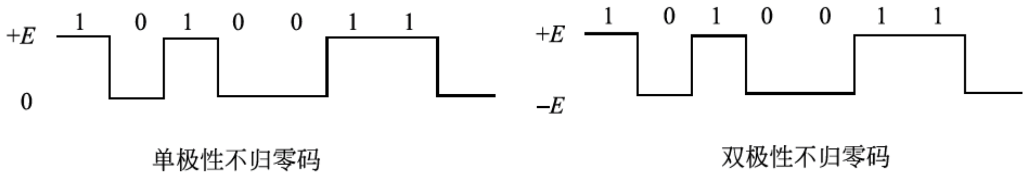


图 2　不归零码

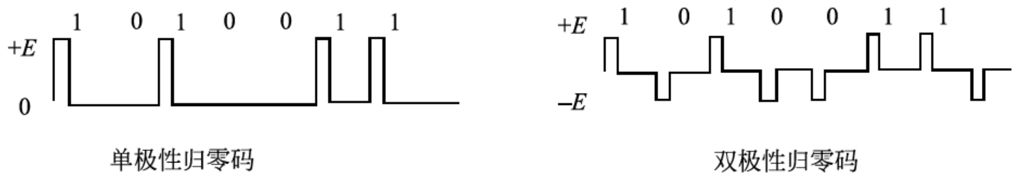


图 3　归零码

双相码又称***曼彻斯特码***，编码规则是用一个周期的正负对称方波表示“0”，而用其反相波形表示“1”。“0”码用“01”两位码表示，“1”码用“10 ”两位码表示。传号反转码的编码规则是：“1”码交替用“11”和“00”两位码表示；“0”码固定地用“01”表示。如下所示：

消息码　1　1　0　0　1　0　1

双相码　10　10　01　01　10　01　10

传号反转码　11　00　01　01　11　01　00

不归零码、归零码、传号交替反转码、双相码及传号反转码的对比关系图如图 4 所示。

实际上，组成基带信号的单个码元波形并非一定是矩形的。根据实际的需要，还可有多种多样的波形形式，比如升余弦脉冲、高斯形脉冲等。

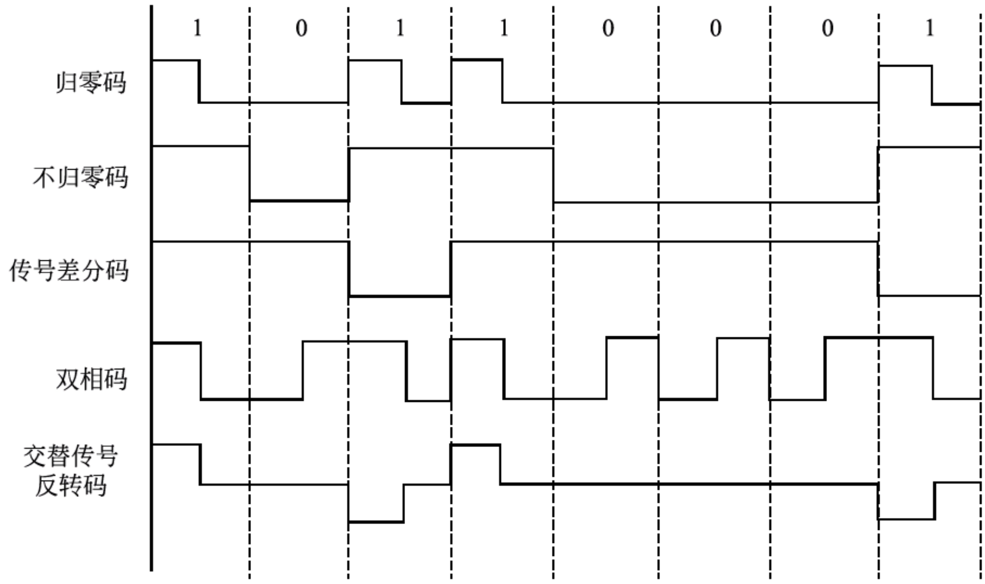


图 4　*五种码形对比图*

在计算机局域网传输中，大量采用基带传输技术。

参考文献

1. ［日］关清三. 数字调制解调基础. 北京：科学出版社，2002

2. 蒋青，于秀兰. 通信原理. 北京：人民邮电出版社，2008

（*董守斌　张凌*）

jishu

基数（cardinal number）　有穷集合元素个数概念在无穷集合上的推广。

空集的基数规定为“0”，非空有穷集合的基数就是它的元素个数，并称它们为有穷基数。无穷集合的基数称为***超穷基数***（或无穷基数）。无穷集合的基数，其意义在于比较两个集合的大小。

集合 A 的基数，用 $\overline{A}$ 或 $\#A$ 表示。若存在从集合 A 到集合 B 的双射，则称 A 与 B 对等（或等势），记为 $A \sim B$。若 A 与 B 对等，则 $\overline{A}=\overline{B}$。

自然数集合 $\mathbf{N}$ 的基数用 $\aleph_0$ 表示，与实数区间 $(0,1)$ 对等的集合称为***连续统***，它的基数用 $\aleph$ 表示。G. Cantor 首先证明 $\aleph_0 \neq \aleph$。

因为从 $\mathbf{N}\times\mathbf{N}$ 到 $\mathbf{N}$ 有如下的双射 π：

$$\pi(n,m)=\frac{(n+m)(n+m+1)}{2}+n,\quad n,\ m\in\mathbf{N}$$

所以 $\overline{\overline{\mathbf{N}\times\mathbf{N}}}=\overline{\overline{\mathbf{N}}}=\aleph_0$，即 $\mathbf{N}\times\mathbf{N}$ 为可数集。

设 A 和 B 为任意两个集合。若有 $B'\subseteq B$ 使 $A\sim B'$，则称 A 的基数小于等于 B 的基数，记为 $\overline{\overline{A}}\leqslant\overline{\overline{B}}$。如果 $\overline{\overline{A}}\leqslant\overline{\overline{B}}$ 且 $\overline{\overline{B}}\neq\overline{\overline{A}}$，则称 A 的基数小于 B 的基数，记为 $\overline{\overline{A}}<\overline{\overline{B}}$。

定理（伯恩斯坦定理）　设 A 和 B 为两集合。若 $\overline{\overline{A}}\leqslant\overline{\overline{B}}$ 且 $\overline{\overline{B}}\leqslant\overline{\overline{A}}$，则 $\overline{\overline{A}}=\overline{\overline{B}}$。

定理（康托尔定理）　若 A 为集合，则 $\overline{\overline{A}}<\overline{\overline{P(A)}}$，其中 $P(A)$ 为 A 的幂集。

以上两定理的证明见参考文献。

设 A 和 B 为两个集合，用 B^A 表示集合 $\{f|f:A\to B\}$。

基数加法　若 $A\cap B=\varnothing$，则称 $\overline{\overline{A\cup B}}$ 为 $\overline{\overline{A}}$ 与 $\overline{\overline{B}}$ 之和，记为 $\overline{\overline{A}}+\overline{\overline{B}}$。

基数乘法　称 $\overline{\overline{A\times B}}$ 为 $\overline{\overline{A}}$ 与 $\overline{\overline{B}}$ 之积，记为 $\overline{\overline{A}}\cdot\overline{\overline{B}}$。

基数的幂　称 $\overline{\overline{B^A}}$ 为 $\overline{\overline{B}}$ 的 $\overline{\overline{A}}$ 次幂，记为 $(\overline{\overline{B}})^{\overline{\overline{A}}}$。

有以下关于基数运算的重要公式:

(1) 若 $n\in\mathbf{N}$ 且 $n\neq0$,则

$\aleph_0+n=\aleph_0=n+\aleph_0$

$\aleph_0+\aleph_0=\aleph_0=\aleph_0\cdot\aleph_0$

$n\cdot\aleph_0=\aleph_0=\aleph_0\cdot n$

(2) 若 $n\in\mathbf{N}$ 且 $n\neq0$,则

$\aleph+n=\aleph=n+\aleph$

$\aleph+\aleph_0=\aleph=\aleph_0+\aleph$

$\aleph+\aleph=\aleph=\aleph\cdot\aleph$

$n\cdot\aleph=\aleph=\aleph\cdot n$

$\aleph_0\cdot\aleph=\aleph=\aleph\cdot\aleph_0$

(3) $\aleph=2^{\aleph_0}$

(4) 若 α,β 和 γ 都是基数,则

$$\begin{cases}\alpha+\beta=\beta+\alpha\\ \alpha\cdot\beta=\beta\cdot\alpha\end{cases}\qquad(\text{交换律})$$

$$\begin{cases}(\alpha+\beta)+\gamma=\alpha+(\beta+\gamma)\\ (\alpha\cdot\beta)\cdot\gamma=\alpha\cdot(\beta\cdot\gamma)\end{cases}\qquad(\text{结合律})$$

$$\begin{cases}\alpha\cdot(\beta+\gamma)=\alpha\cdot\beta+\alpha\cdot\gamma\\ (\alpha+\beta)\cdot\gamma=\alpha\cdot\gamma+\beta\cdot\gamma\end{cases}\qquad(\text{分配律})$$

$$\begin{cases}\alpha^{\beta+\gamma}=\alpha^{\beta}\cdot\alpha^{\gamma}\\ (\alpha\cdot\beta)^{\gamma}=\alpha^{\gamma}\cdot\beta^{\gamma}\\ (\alpha^{\beta})^{\gamma}=\alpha^{\beta\cdot\gamma}\end{cases}\qquad(\text{指数定律})$$

$$\begin{cases}0\cdot\alpha=0=\alpha\cdot0\\ 0+\alpha=\alpha=\alpha+0\\ \alpha^{0}=1\end{cases}$$

$$\begin{cases}1\cdot\alpha=\alpha=\alpha\cdot1\\ \alpha^{1}=\alpha\\ 1^{\alpha}=1\end{cases}$$

(5) 若 α 为超穷基数且 β 为非零基数,则

$\alpha+\beta=\max\{\alpha,\beta\}$

$\alpha\cdot\beta=\max\{\alpha,\beta\}$

参考文献

王兵山,王长英,周贤林,等. 离散数学. 长沙:国防科技大学出版社,1985

(王兵山)

jiyu anli de tuili

基于案例的推理(case-base reasoning, CBR) 依据过去求解类似问题所积累的案例或经验来求解新问题的一种推理模式,是知识系统中的一种重要推理技术。譬如,一个较简单的案例可以是一组特征和对应的结论,一个复杂案例可能是由一个案例及其若干子案例组成的层次结构,层次结构中的子案例可能又有若干子案例,等等。

CBR研究起源于认知科学,其思想最早由R. Schank在1982年出版的《动态记忆》(*Dynamic Memory*)中提出。之后,J. Kolodner等人开始在计算机上实现CBR系统。进入20世纪90年代,CBR越来越受到关注,并得到深入研究和较广泛的应用。

基于案例的推理较符合人类记忆和推理的方式,是一种具有发展前景的问题求解方法。

基于案例推理的基本方法

一般认为,CBR的推理过程是一个4R(Retrieve、Reuse、Revise、Retain)循环,即案例的检索、重用、修订和保留。

在基于案例推理的问题求解方法中,人们将过去对典型问题的求解案例,按一定的表达方式存储起来,组织成案例库。当用户求解某一新问题时,以案例库中案例的组织方式输入待求解的问题,即待求解案例。CBR系统根据问题的描述,利用案例检索机制(Retrieve),从案例库中寻找与待求解案例匹配或近似匹配于待求解案例的案例。如果找到的案例符合问题求解的要求,则将其作为对问题的解输出(Reuse);否则,根据待求解问题描述,对检索出的相近案例进行修改(Revise),以产生一个符合问题求解要求的解,将其输出。同时将这个解作为一个新的案例再存储到案例库中,以备将来使用(Retain)。

与基于规则的方法相比,CBR不需要详尽的领域知识模型,通过运行不断积累案例,丰富案例库,使系统性能不断改善。

按任务类型可将CBR分成解释型和问题求解型。解释型利用以往的案例对新情况进行分类或特征化,如司法领域的案例推理;问题求解型利用先前的案例为新问题提出解决方案,可用于计划、设计、诊断等领域。

案例的组织

案例的组织主要有最近邻法、归纳索引法和知识导引法等。①最近邻法　基本思想是将案例间的某种距离作为案例组织的标准。常见的最近邻组织方法是将表征案例的 n 维特征向量看成 n 维空间的一个点,然后按某种 n 维空间距离对所有的点进行排序;或者对案例的特征的权值求和,再根据该和数组织案例。②归纳索引法　通常有两种:一种是分类网方法,将案例按“抽象/具体”的层次结构加以组织,上层存放抽象程度高的案例,下层存放抽象程度低(或更为具体)的案例;另一种是决策树方法,根据案例特征向量在不同维上的信息差异,将案例组成决策树。③知识引导法　一种启发式的案例组

织方法。在案例库中,系统按目前已知的索引知识判断案例特征的重要性,并按案例特征的重要性组织案例库。

案例检索与改写

目标是快速地从案例库中找到与待解问题相同或最相似的案例集。案例检索的效率与案例组织方法密切相关:①按最近邻法组织的案例库　依据定义的某种距离,查找与待解问题距离最小的案例;②按归纳索引法组织的案例库　应采用自顶向下,逐层求精的策略,搜索的层数越大,其相似程度越高;③按知识索引法组织的案例库　一种简单的方法是对案例的各种特征加权,对重要的特征优先检索。

案例改写表现为系统的学习功能,主要依据待解问题的特点,利用案例修改策略和案例修改知识,对查找到的最相似案例进行适当调整,使之适合于当前待解问题。

CBR 的应用领域,如信息安全中的入侵检测、生物信息学中的 DNA 序列研究、医学领域的多种疾病的诊断等。

CBR 未来的发展方向:案例表示的新方法的开拓,如本体表示方法;案例库的维护与检索方法,如聚类方法等;案例推理作为**解释机制**的研究,如将 CBR 作为**专家系统**的解释机制。

参考文献

1. Kolodner J, Case-based reasoning. San Francisco: Morgan Kaufmann, 1993

2. Aamodt A, Plaza E. Case-based reasoning: Foundational issues, methodological variations, and system approaches. AI Communications. IOS Press, 1994, 7(1): 39-59

3. Simic D, Kurbalija A, Budimac Z. An application of case-based reasoning in multi dimensional database architecture. Berlin: Springer-Verlag, 2003

(何炎祥　朱福喜)

jiyu bianjie de bingxing tuxiang fenge fangfa

基于边界的并行图像分割方法(boundary-based parallel image segmentation methods)　**图像分割**的一大类方法。这种方法基于区域间像素的灰度不连续性以检测区域边界上的像素,且处理过程中对不同像素的判断和决策是独立和并行做出的。

对灰度图像,边缘总存在于具有不同灰度值的相邻区域之间,所以可利用一阶和二阶导数来检测边缘像素。利用一阶导数的梯度算子**包括**罗伯特交叉算子、蒲瑞维特算子和索贝尔算子。利用二阶导数的算子包括拉普拉斯算子、马尔算子等。

在有噪声时,用各种算子得到的边缘像素常是孤立的或分小段连续的。为组成区域的封闭边界以将不同区域分开,需要将边缘像素连接起来。边界闭合要借助边缘像素间的相似性。用梯度算子对图像处理可得到像素两方面的信息:①梯度的幅度;②梯度的方向。因此如果两个像素邻接且它们的梯度幅度和方向的差值均小于给定的阈值,就可将它们连接起来。如对所有边缘像素都进行这样的判断和连接就有可能得到闭合的轮廓。

参考文献

1. Gonzalez R C, Woods R E. Digital image processing. 3rd ed. Prentice Hall, 2008

2. 章毓晋. 图像工程(中册)—— 图像分析. 3 版. 北京:清华大学出版社,2012

(章毓晋)

jiyu bianjie de chuanxing tuxiang fenge fangfa

基于边界的串行图像分割方法(boundary-based sequential image segmentation methods)　**图像分割**的一大类方法。此类方法先基于图像区域间像素的灰度不连续性以检测区域边界上的像素,再将它们串行连接成闭合边界。由于这里考虑了图像中边界的全局信息,常在图像受噪声影响较大时还能取得较鲁棒的结果。两种典型的边缘检测和边界连接互相结合顺序进行的方法为图搜索和动态规划方法。

近年提出的基于图论的**图割方法**(Graph Cut Method)也是一种基于边界的串行分割方法。它将图像分割问题转化为最优化问题来解决,其主要步骤为(参见图 1):

(1) 将待分割图像 I(图 1(a))映射为一个对弧加权的有向图 G(图 1(b))。即将 I 中每个像素看成 G 中的一个结点,而像素间的邻接关系用 G 中的弧来表示。

(2) 确定目标种子 O 和背景种子 B(图 1(a)),并针对它们构建两个特殊的图结点:源结点 S 和汇结点 T(图 1(b))。这里所确定的目标种子和背景

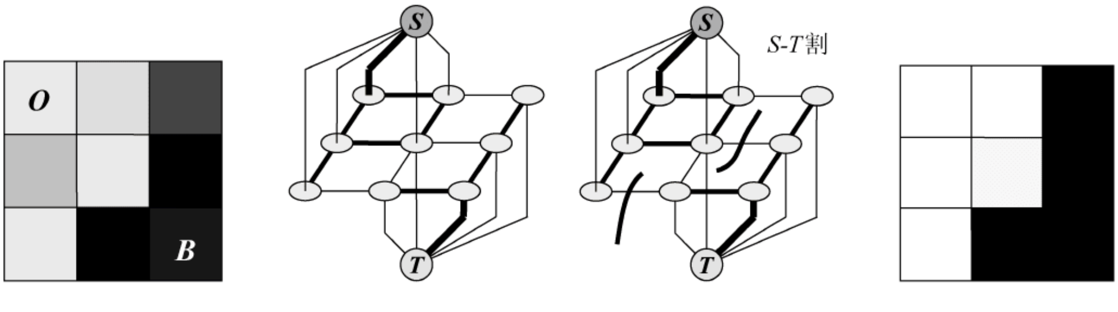


图 1　图割方法的简单示意图

种子应分别是最终分割结果中目标或背景的一部分。

(3) 对 G 中的各个弧赋予一定的代价。在 G 中的一个割将结点分成两组，其代价是这个割所穿过/跨越的弧的代价之和。代价最小的割称为最小 S-T 割(图(c))，将结点分成两组不重叠的子集。

(4) 使用最大流图优化算法来确定对 G 的图割，从而区分目标和背景像素的结点，得到分割结果(图 1(d))。

参考文献

1. Sonka M, Hlavac V, Boyle R. Image processing, analysis, and machine vision. 3rd ed. Thomson, 2008

2. 章毓晋. 图像工程(中册)—— 图像分析. 3 版. 北京：清华大学出版社，2012

(章毓晋)

jiyu goujian de ruanjian kaifa fangfa

基于构件的软件开发方法(component-based software development method，CBSD)　一种基于预先开发好的**软件构件**，通过将其集成组装的方式来开发软件系统的方法。又称基于构件的软件工程(CBSE)，它是**软件复用**的实现方式之一。其根本目的仍然是为了提高软件开发的质量和效率。

基于构件的软件开发方法(CBSD)的兴起主要是源于如下不同的背景：一是在学术研究方面对现代软件工程思想，特别是对软件复用技术的高度重视；二是在技术研发方面所取得的有效进展，如，虽然缺少理论的支持，但在图形用户界面(GUI)和**数据库**应用中基于部件的组装技术的成功应用；三是一些主流互操作技术开发者的积极推动，如 OMG 的 CORBA/CCM、微软公司的 COM/DCOM 以及 SUN 公司的 EJB 已成为主流的构件实现规范，相应的**软件中间件**平台规范也已获得较为普遍的接受；四是由于面向对象技术的广泛使用，提供了构件和使用构件的概念基础和实用工具，事实上，主流的构件实现模型均基于对象技术。

从开发方法的角度来看，CBSD 提供了一种自底向上的、基于预先定制包装好的**类**属元素(构件)来构作应用系统的途径。应该看到，CBSD 的发展和中间件技术的发展是密切相关的，正是中间件技术及其平台提供了构件开发和构件组装的技术基础和机制。因此，当前 CBSD 讨论的重点主要局限于基于 COM/DCOM、CORBA/CCM 和 EJB 等主流规范的二进制级构件。

从复用的角度看，CBSD 支持的是黑盒、组装式复用方式。系统开发者不能对构件进行源代码级的修改，最多只能是通过参数方式进行适应性调整。构件的组装在中间件平台上进行，构件间通过中间件提供的通信协议和设施进行交互。

CBSD 的主要活动包括：构件获取、构件认证、构件适应性修改、构件组装和应用部署。

构件获取　根据应用需求，去寻找符合或基本符合需求的构件，通常途径有三，一是来自第三方构件开发商，二是货架商品式的(COTS)构件，三是集成者根据特殊应用需求开发。

构件认证　对候选构件进行有用性、可用性、可信性和复用历史等方面的考察，以确定是否可用于新应用系统中。认证可由集成者自己完成，也可委托第三方认证。

构件适应性修改　由于应用的特殊性，有时需要对候选构件进行适当修改。由于 CBSD 所支持的黑盒复用模式，集成者只能通过参数调整的方式修改构件。如要涉及代码级的修改，通常仍需由原构件开发商完成。

构件组装　根据应用的功能及非功能需求，将相关的构件组装在一起，主要工作涉及一些胶合代

码和构件间连接关系描述信息的生成。

应用部署 将构成应用系统的各构件或构件组分别部署到相应的运行环境中，也即是将组装形成的应用系统“安装”到相应中间件平台上。部署完毕，应用系统方可投入运行。

基于构件的软件开发方法（CBSD）正在受到越来越多的关注，然而，CBSD 的有效性和高效性还面临一些挑战。CBSD 当前主要着重于目标码层次的互操作能力，缺少涉及需求和设计层次的系统化方法学支持，因此，需要和软件复用的已有研究成果进行无缝整合。同时，大量可供使用的构件的存在是 CBSD 能够被广泛应用的前提，而这正是目前大多数应用领域所欠缺的。此外，多种构件实现规范及体系结构风格的共存，使得软件体系结构失配成为 CBSD 方法的一个难题。

参考文献

1. Heineman G T, Councill W T. Component-based software engineering: putting the pieces together. Reading, MA: Addison-Wesley, 2001

2. Mili H, Mili A, Yacoub S. Reuse based software engineering: techniques, organizations, and controls. John Wiley & Sons, 2002

3. Hong Mei, Jichuan Chang, Fuqing Yang. Software component composition based on ADL and middleware. Science in China (F), 2001, 44(2): 136-151

（谢涛　梅宏）

jiyu neirong de tuxiang jiansuo

基于内容的图像检索（content-based image retrieval, CBIR） 根据图像内容的相似性从图像数据库中查找出与用户需求相似的图像的一种技术。与传统的基于文本的图像检索方式不同，CBIR 不需要对数据库中的图像进行关键字、标题等人工文字标注，而直接从图像本身提取内容线索，使得检索过程更加有效，适应性更强。CBIR 涉及**数字图像处理**、**计算机视觉**、**数据挖掘**、**机器学习**、**模式识别**、人机交互等诸多学科领域。

CBIR 中的主要问题是解决从图像中提取的特征与用户所需求的图像内容间的“语义鸿沟”问题。依据语义层次的高低，用户的检索类型可分为三个层次：①检索具有特定颜色、形状、纹理等低层特征的图像，如“找一幅蓝色占 60% 的图像”；②检索包含特定类型的物体或场景的图像，如“找一幅天空中有飞机的图像”；③检索蕴含特定意义或意图的图像，如“找一幅描述高兴的图像”。上述三个层次的语义由低到高，其中第一个层次的检索比较直接（但通常只适用于待检索图像内容可直接由低层图像特征来表征的特定应用环境），通过提取相关特征并计算与目标特征的相似度即可完成；后面两个层次的检索需要通过尽可能地利用关于待检索内容的先验知识来解决语义鸿沟问题，如需要事先训练特定类型的物体检测器或人脸表情分类器（可采用 Adaboost、SVM 等来机器学习方法来实现）、对待检索场景中物体间的共存与空间拓扑等关系进行分析与建模等。

图像检索的性能需要有一个客观全面的性能评价标准。比较通用的性能评价准则是有效性（检索结果正确与否）、效率（检索速度）和灵活性（对不同应用的适应性）。目前对检索性能的评价更多地注重其有效性，可采用如查全率、查准率等来评测。为提高图像检索的性能，可以考虑综合多种特征的图像检索以及考虑图像检索中的特征优化问题。此外，引入相关反馈技术也是提高检索性能的一个行之有效的方法。其目标是从用户与查询系统的实际交互过程中进行学习，发现并捕捉用户的实际查询意图，并以此修正系统的查询策略，从而得到与用户实际需求尽可能相吻合的查询结果。

CBIR 技术将对大规模图像信息的管理和访问提供有力支持。它可以广泛应用于信息检索服务、犯罪预防、医疗诊断、新闻和广告、商标和知识产权、地理信息和远程遥感、教育培训和军事等领域。

参考文献

1. 周明全，耿国华，韦娜. 基于内容图像检索技术. 北京：清华大学出版社，2007

2. 孙君顶，赵珊. 图像低层特征提取与检索技术. 北京：电子工业出版社，2009

（郄慧军）

jiyu quyu de bingxing tuxiang fenge fangfa

基于区域的并行图像分割方法（region-based parallel image segmentation methods） **图像分割**的一大类方法。此类方法基于图像区域内像素的灰度相似性以检测区域内的像素，且该过程对不同像素的判断和决定是独立和并行地做出的。

取阈值技术是最广泛使用的基于区域的并行分割方法（其他还有像素分类、模板匹配等）。

最常用的取阈值分割模型可描述如下：假设图

像由具有单峰灰度分布的目标和背景组成,在目标或背景内部的相邻像素间的灰度值是高度相关的,但在目标和背景交界处两边的像素在灰度值上有很大的差别。满足这些条件的图像的灰度直方图基本上可看作是由分别对应目标和背景的两个单峰直方图混合而成。如果这两个分布大小(数量)接近且均值相距足够远,而且均方差也足够小,则直方图应是双峰的。对这类图像常可用取阈值方法来较好地分割。

最简单的利用取阈值方法来分割灰度图像的步骤如下。首先对一幅灰度取值在 $g_{\min}$ 和 $g_{\max}$ 之间的图像确定一个灰度阈值 $T(g_{\min} < T < g_{\max})$,然后将图像中每个像素的灰度值与阈值 T 相比较,并将对应的像素根据比较结果(分割)划为两类,即

$$g(x,y) = \begin{cases} 1, & \text{如 } f(x,y) > T \\ 0 & \text{如 } f(x,y) \leq T \end{cases}$$

由上述讨论可知,取阈值分割方法的关键问题是选取合适的阈值。阈值一般可写成如下形式

$$T = T[x,y,f(x,y),q(x,y)]$$

式中,$f(x, y)$代表像素点(x, y)处的灰度值;$q(x, y)$是该点邻域的某种局部性质。借助上式,可对取阈值分割方法分类,所对应的阈值有 3 种。

(1) 全局阈值　仅根据各个像素的本身性质 $f(x,y)$来选取而得到的阈值。典型的方法包括极小值点阈值法、最优阈值法等。

(2) 局部阈值　根据像素的本身性质 $f(x,y)$ 和像素邻域性质 $q(x,y)$ 来选取得到的阈值。典型的方法包括直方图变换法、灰度-梯度散射图法等。

(3) 动态阈值　根据像素的本身性质 $f(x,y)$,像素邻域性质 $q(x,y)$ 和像素位置坐标(x,y)来选取得到的阈值(与此对应,可将前两种阈值称为固定阈值)。

参考文献

1. Zhang Y J. Image engineering: processing, analysis, and understanding. Cengage Learning, Singapore. 2009

2. 章毓晋. 图像工程(中册)——图像分析. 3 版. 北京:清华大学出版社,2012

（章毓晋）

jiyu quyu de chuanxing tuxiang fenge fangfa

基于区域的串行图像分割方法(region-based sequential image segmentation methods)　**图像分割**的一大类方法,此类方法基于图像区域内像素的相似特性来检测区域内的像素,其过程对不同像素的判断是串行进行的,必要时还可借助反馈迭代进行。

串行分割方法中进行的判断要根据一定的准则来进行。一般来说如果准则是基于图像灰度特性的,则该方法可用于灰度图像的分割;如果准则是基于图像的其他特性(如纹理)的,则该方法也可用于相应特性图像的分割。常用的两种技术是区域生长和分裂合并。区域生长的基本思想是将具有相似性质的像素集合起来构成区域。具体步骤是:先对每个需要分割的区域找一个种子像素作为生长的起点,然后将种子像素周围邻域中与种子像素有相同或相似性质的像素合并到种子像素所在的区域中。将这些新像素作为新的种子像素继续进行上面的过程,直到再没有满足条件的像素可被包括进来。分裂合并的基本思想是先把图像分成任意大小且不重叠的区域,然后再合并或分裂这些区域来实现图像分割。

参考文献

1. Gonzalez R C, Woods R E. Digital image processing. 3rd ed. Prentice Hall, 2008

2. 章毓晋. 图像工程(中册)—— 图像分析. 3 版. 北京:清华大学出版社,2012

（章毓晋）

jiyu tuxiang de huizhi

基于图像的绘制(image-based rendering, IBR)　以图像作为主要基元代替传统的几何基元来绘制合成新视图的技术,也称基于图像的建模和绘制(image-based modeling and rendering)。因为强调以图像为中心,而不像传统图形学以几何为中心,所以人们习惯称之为基于图像的绘制(IBR)。IBR 与基于几何的计算机图形绘制方法不同。首先,IBR 利用一组真实场景照片来生成场景的新视图,故能反映极为丰富的景物细节和色彩,有效地克服了基于几何的图形学存在的建模复杂度高和绘制真实感不足的缺陷;其次,IBR 的绘制速度不依赖于场景复杂度,有效地克服了基于几何的图形学绘制速度随场景模型复杂度增加而下降的缺陷。IBR 技术为实时绘制极为复杂的真实场景提供了可能。

基于图像的绘制技术始于 20 世纪 90 年代初。有关 IBR 的研究内容包括:①数据采样方法,即如何获得场景图像,以及如何以最少的图像采样满足反走样的绘制结果。②表示和绘制方法,即基于图像的场景表示和视图合成技术。这是 IBR 的重点研究内容。根据是否使用几何信息和使用几何信息

的多少,可以将各种 IBR 技术定位在一条水平轴上,该轴的左端是表示完全基于图像的方法(如 Light Field 和拼图技术),右端表示明确使用几何信息的方法(如 Layered-depth Images 和纹理映射)。如果某一 IBR 方法越少越隐式地使用几何信息,则它的定位就越靠近左端;反之则越靠近右端。例如视图变形(View Morphing)就认为是一种隐含使用几何信息的 IBR 方法,它利用**计算机视觉**技术来获得已知视图的对应点,并计算相机的成像矩阵。③数据压缩和传输,IBR 的特点是图像密集型,如何有效地压缩并能快速转换为需要的表示方式进行显示是一项重要的研究内容。

在 IBR 的发展过程中,不仅计算机图形学界的学者对其投入了极大的热情,同时也吸引了许多计算机视觉、图像和信号处理领域学者。可以说 IBR 推动了这些领域的交叉融合。

参考文献

1. Heung-Yeung Shum, Shing-Chow Chan, Sing Bing Kang. Image-Based Rendering,Springer Science + Business Media LLC, 2007

2. 石教英. 虚拟现实基础及实用算法. 北京:科学出版社,2002

3. Levoy M, Hanrahan P. Light field rendering. In: Computer Graphics (SIGGRAPH'96 Conference Proceedings), 2006, 31-42

4. Gortler S J, He L W, Cohen M F. Layered depth images. In: Computer Graphics (SIGGRAPH'98 Conference Proceedings), 2006, 231-242

5. Seitz S M, Dyer C M. View morphing. In: Computer Graphics (SIGGRAPH'96 Conference Proceedings), 2006, 21-30

(徐丹)

jiyu tuxiang de zaoxing

基于图像的造型(image-based modeling, IBM) 指对实际对象或场景进行拍摄,根据获得的一幅或一组图像来重建其三维几何和外观模型的过程和技术。传统计算机图形学采用几何表示和纹理合成来建立对象或场景的模型,称为基于几何的造型技术。建立一个复杂对象或场景的三维几何模型是一项既困难又十分耗时的工作,迄今仍缺乏有效的解析方法来综合复杂的外观特征。基于图像的造型技术直接用包含丰富细节和逼真外观特征的采样图像来造型,改善了模型的复杂度和真实感。同时,基于图像造型技术的造型过程自动化程度高,降低了造型成本。基于图像的造型技术是**计算机图形学与计算机视觉**两个学科交叉的产物,在 20 世纪 90 年代中期兴起,一直受到学术界重视,成为研究、开发和应用的热点。

基于图像的造型技术分为两类:一类称为主动技术,另一类称为被动技术。主动技术采用人为控制照明的方法进行拍摄,以获得具有特殊效果的图像。例如将具有已知结构的光(常用的有光栅条纹,或仅仅一条棒形阴影)投射到对象上进行拍摄,根据光栅条纹在图像上的扭曲形状来恢复对象表面的三维模型。被动技术在拍摄时对光照条件不加任何限制,具有较高的灵活性,完全根据对象自身结构来重构模型,其重建技术相对比较复杂。被动技术包括多种方法,它们之间的主要区别在于对照相机校准的要求不同,及重建过程中人机交互量的多少(即自动化程度的高低)两个方面。常用的被动技术有以下几种方法:

(1) 基于立体视觉的图像造型方法　假如用符合针孔模型的照相机在不同视点上拍摄对象,得到多幅图像。在多幅图像上提取一系列特征点,并找出它们的匹配关系,据此进行相机自标定,恢复相机的各种(内在的和外在的)参数。然后将两幅相关图像重投影到某个公共区域上,以使对应极线对齐,再通过配准得到对应的立体像对。根据立体像对与相关视线关系恢复像素点的相对深度,达到三维重建的目标。

(2) 基于可见轮廓的图像造型方法　假设从不同视角拍摄同一个物体,得到一系列图像。从每幅图像上可以得到从不同视角看到的物体的轮廓线。物体的轮廓线是理解物体几何形状的一个重要线索。从每个视点发出经过各自轮廓点的射线分别构成一系列锥形外壳。这些不同角度的锥壳的交集构成了物体的可见外壳(三维模型)。该方法特别适用于带高光的、透明的、半透明的和带绒毛的物体的造型。

(3) 基于表面光场的图像造型方法　光场是指某一点上朝给定方向发出的光线。这样,图像上的像素可以看作是物体表面光场中朝视点方向发出的光线集合。该方法对给定物体的上半空间(假定下半空间为不可见)进行全方位密集采样,得到成百上千帧图像。这些图像记录了物体在上半空间的表面光场数据。当要求再现上半空间某视角方向上物体的外观时,只要对物体的表面光场数据按给定的视点坐标和视线方向进行重采样,即从表面光场选

取符合条件的光线构成物体在给定视角上的外观图像。这是一种用于获取物体外观模型的方法，不能获取物体的几何模型。由于该方法采样数据量极大，因此有效的光场数据压缩技术成为其关键技术。

有关基于图像的造型的主要研究内容为：①采样方法研究；②采样数据的紧凑表示（压缩技术）；③有效的绘制方法等。

参考文献

1. Long Quan. Image-based modeling. Springer, 2010

2. 石教英. 虚拟现实基础及实用算法. 北京：科学出版社，2002

（石教英　刘利刚）

jiyu wuli de donghua

基于物理的动画（physically-based animation） 将物理特性引入模型，并允许对模型的行为进行数值模拟的**计算机动画**技术。它可应用物理定律及基于约束的技术来推导和计算物体随时间运动和变化的状况。

这种技术的特点如下：

（1）采用**基于物理的造型**，使模型中不仅包含几何信息，也包含物理信息。它将与行为有关的物理特性（如质量、力、力矩、惯量等）、形体间的约束关系及其他与行为的数值模拟相关的信息引入模型中，甚至将图像绘制技术与该模型的描述相结合。这样就模糊了通常的造型、动画、图像绘制之间的界限，改变了计算机动画通常采用的造型—动画—图像绘制相分离的流水线机制。同时，在实现形体的动态变化时也不再需要由人工详细规定或调整形体的几何数据和拓扑结构。

（2）在形体的运动和变化的控制过程中引进了物理定律。用这种方法控制的运动更符合实际状况，更加自然也更有吸引力。

真实感效果是计算机动画所追求的目标之一。它不仅体现在“照相逼真”上，而且还体现在动画形体的造型及其动态变化和运动的逼真模拟上。基于物理的动画正是为解决这一问题而提出的。

计算机动画中要解决的另一个问题是如何正确地、容易地控制动画中形体的运动和变化。传统的**关键帧动画**在控制层次上偏低，需要由动画设计者直接控制形体的位置等相关参数，这对复杂的运动来说是相当烦琐和困难的，有时甚至是不可能的。

基于物理的动画采用基于物理模型的模拟技术，使动画设计者通过对形体及影响形体动作的环境随时间而变化的描述来实现控制，这就提供了很大方便，即使是初学者也可自动地生成真实的运动。基于物理的动画已发展成计算机动画中的重要课题。它与机器人学、**人工智能**、面向对象的方法相结合，适用于通用的集成环境中，实现任务级上的运动控制和描述。如能有效地解决运动和变化的计算速度问题，必将在科学计算可视化、视觉模拟、**人体动画**等领域发挥更大的作用。

参考文献

1. Parent R. Computer animation：algorithms and techniques. 2nd ed. San Fransisco，CA：Morgan Kaufmann，2007

2. Kerlow IV. The art of 3D computer animation and effects. 4th ed. Wiley Publishing，2009

（王裕国）

jiyu wuli de zaoxing

基于物理的造型（physically-based modeling） 在**几何造型**的基础上，将形体的物理规则引入其模型，并对其行为和形状的变化进行模拟的造型技术。

许多物体的行为和形状是由该物体与其他物体的相互关系及有关的物理性质所决定的。例如悬挂在两根柱上的链条呈弧状下垂，它是由重力及维持链条间的连接力所决定的；天空包括云、雾、霭、晕轮，涉及复杂的大气散射、折射、衍射、衰减等物理机理。基于物理的造型就是解决这类物体（可以是刚体、弹塑体、流体，也可以是人、动物、机器人的关节体等）的具有物理真实性的动态造型问题的技术。它与传统的造型技术不同，在建造模型时，不仅包含几何信息，也引入了导致物体形状和行为变化所需要的其他信息，即物体的物理性质和动态模拟机制。这些信息涉及运动学（位置、速度等）、动力学（力、力矩、质量、能量等）以及物体内或物体间的各种约束（位置或方位约束、动力学约束、能量约束）等。一般用粒子系统、连续体或刚体的受约束动力学、流体力学原理来描述。将物理模型引入后需要进行必要的简化加速，最后表现物体的形体和动态变化。除这些用于确定物体运动和变化的特性外，也可将相关图像绘制技术应用于对模型的描述中。

基于物理的造型的控制是一个重要问题。如果模型的行为是模型所固有的，那么模型将按照它自己的规律进行动态变化。在数学上，模型的行为变化常可用微分方程组描述，用户除给定微分方程组的初值外，很少有其他的控制手段。如果想对模型

的行为有更多的控制，必将对行为的真实性产生一定影响。必须在这两者之间找到一个合理的折中。目前基于物理的造型的控制多采用基于约束的技术，允许用户规定一个约束集，要求模型的行为满足这些约束。当有多个约束时，应给出优先级，以使最重要的约束首先被满足。约束的规定是复杂的，某些约束可由一组数学等式或不等式给定，但更多的是采用微分方程组来描述。

基于物理的造型是涉及**计算机图形学**、应用数学、物理学、计算力学、光学等多种学科的技术。当前，它主要应用于**计算机动画**、**计算机游戏**、影视特技、军事仿真、建筑景观设计、**虚拟现实**等领域。

参考文献

1. Foley J D, van Dam A, et al. Computer graphics: principles and practice. 2nd ed. Addision Wesley, 1995

2. Barzel R, Barr A H. Physically based modeling for computer graphics: a structured approach. Morgan Kaufmann, 1992

3. Pharr M, Humphreys G. Physically based rendering: from theory to implementation. 2nd ed. Morgan Kaufmann, 2010

（王长波）

jizhun chengxu

基准程序（benchmark programs）

一类用于对计算机系统的性能指标等进行定量的和可对比的测试的程序，试图提供一个客观、公正的评价机器性能的标准。一组标准的测试程序要提供一组控制测试条件和步骤的规范说明，包括测试平台环境、输入数据、输出结果和性能指标等。

不同基准测试程序的侧重目的不同，有的测试CPU性能、有的测试文件服务器性能、有的测试输入输出界面、有的测试网络通信速度等。根据不同用途，测试程序可有专用和通用之分。大部分常见的基准测试程序如下。

Whetstone 基准程序 一组用于测试系统性能的综合型基准测试程序，最早由英国国家物理实验室（NPL）开发。它包括浮点运算和整数运算，涉及数组下标索引、子程序调用、参数传递、条件转移和超越函数等。Whetstone 规模不大，对存储器容量要求小且较多使用高速缓冲存储器，因而适用于评估小型的科学、工程应用系统。除了可以测试机器的硬件性能外，还可以用来评估系统数学程序集、语言编译器及其处理效率。测试结果用 KWIPS（每秒执行 1000 条 Whetstone 指令）或 MWIPS（每秒执行 1 000 000 条Whetstone 指令）表示。

Dhrystone 基准程序 一组用于测量编译系统和计算机性能的基准测试程序。最初以 Ada 语言写成并公布于 1984 年，目前主要用 C 语言版本。与 Whetstone 程序不同，Dhrystone 程序中没有浮点指令。它是 CPU 密集型测试程序，由很多整型语句和逻辑语句的小循环组成，主要用于测量整数与逻辑运算的性能，着重反映操作系统、编译器等系统软件的程序特点。测试结果用每秒计算多少次 Dhrystone（每秒完成多少个 Dhrystone 循环）来反映机器的性能，现在通常把 VAX-11/780 机运行 Dhrystone 程序的测试结果 1757 Dhrystones/s 作为 1，其他机器运行 Dhrystone 程序的结果与它作比较，Dhrystone 值越大性能越好。

ScaLAPACK 测试程序 ScaLAPACK 是线性代数运算程序包 LAPACK 在分布式存储环境中的扩展，是美国能源部 ODE2000 工程支持开发的 20 多个 ACTS（Advanced Computing Test & Simulation Program）工具箱之一。主要运行在基于分布式存储和消息传递机制的 MIMD 计算机以及支持 PVM 或 MPI 的机群上。ScaLAPACK 测试程序一共有 18 个，分别进行 LU、Cholesky、带状 LU、带状 Cholesky、普通三对角、带状三对角、QR（RQ、LQ、QL、QP 和 TZ）、线性最小二乘、上 Hessenberg 化简、三对角化简、双对角化简、矩阵求逆、对称特征值问题、广义对称特征值问题、非对称特征值问题和奇异值问题的计算。

LMbench 测试程序 一套为 UNIX 和 POSIX 制定的可移植且符合 ANSI C 标准的微型性能测评工具。它主要衡量系统的两个关键特征：带宽和反应时间，旨在使系统开发者深入了解关键操作的基础成本。LMbench 是符合 GNU 的 GPL 版权标准的开源软件，代码量较小且容易扩展，可以按常规组合成不同的形式以测试其他内容。

STREAM 测试程序 一款综合性的持续内存带宽测试程序，能够对单环境和多重负荷时的内存性能进行测评，是目前高性能计算系统测试的重要指标，测试结果以 MB/秒来衡量。

NPB 测试程序 美国国家航空航天局（NASA）开发的一套代表流体动力学计算的应用程序集，它已经成为公认的用于评测大规模并行机和超级计算机的标准并行测试程序。NPB 主要分为 NPB1、NPB2、NPB3 三个版本，其中 NPB1 由 8 个程序组成，测试范围从整数排序到复杂的数值计算，包括 5 个

核心程序(EP、FT、MG、CG 和 IS)和 3 个模拟应用(LU、SP 和 BT),反映了航空、物理学应用高性能并行计算的全貌。NPB2 则是对 NPB1 进行补充后提供的 MPI 版本,增加了 D 规模测试集,相比于 C 规模测试集,数据集扩大了约 16 倍,负载扩大了约 20 倍。NPB3 进一步增加了 OpenMP、HPF 和 Java 等支持。NPB 中每个基准测试程序有 5 类问题规模,分别为 A、B、C、S 和 W。其中,A 类的规模最小,C 类的规模最大,而 W(Workstation)类通常用于工作站,S(Sample)类是样例程序。

PARKBENCH 测试程序　一组用于评价大型可扩展系统的科学计算性能的并行测试程序,重点放在可扩展、分布存储、消息传递的体系结构上,是在 1992 年超级计算国际会议上确定的项目。主要目标是确定并行机用户和厂商双方都能接受的一批并行测试程序及标准。PARKBENCH 很庞大,用于评价计算机系统支持各种具有不同需求的科学计算应用的性能,目前包括底层基准程序、核心基准程序、压缩应用基准程序、HPF 编译基准程序等。

STAP 测试程序　一套用于实时雷达信号处理系统的测试程序。STAP 程序最初由美国麻省理工学院的林肯实验室开发,后来在南加州大学被转换成并行 STAP,用来评估各种大规模并行处理计算机。STAP 程序是密集计算,要求在不到 1 s 时间内对 O(102-104)MB 的数据完成 O(1010-1014)的浮点操作,由 APT 自适应处理试验台、HO-PD 高阶后多普勒、BM-Stag、EL-Stag 和 GEN 五个程序组成。

Linpack 基准程序　一套用以评测计算机计算性能的通用数学程序集,主要功能是解线性方程组和线性最小二乘问题,使用基础线性代数子程序库 BLAS 完成基本的向量和矩阵运算,最早的矩阵大小为 100×100,300×300,后来以 1000×1000 为标准。随着 MPP、CC-NUMA、Cluster 等大型并行计算机的出现,不再限定矩阵的规模。HPL 是 Linpack 针对现代并行计算机提出的测试方式,用户在不修改任意测试程序的基础上,可以调节问题规模大小(矩阵大小)、使用到的 CPU 数目、使用各种优化方法等来执行该测试程序,以获取最佳的性能。Linpack 的结果按每秒浮点运算次数表示。

SPEC 基准程序　由标准性能评估机构开发的一组用于计算机性能综合评测的程序,旨在确立、修改以及认定一系列服务器应用性能评估的标准。1989 年 SPEC 提出了 1 组共 10 套基准测试程序,其中用于测量机器整数运算性能的程序 4 套,即 SPECint89,用于测试浮点数运算性能的程序共 6 套,即 SPECfp89;后来 SPEC 又推出 SPEC92,SPEC95,SPEC2000 等。SPEC 现已推广至多个应用领域,并推出了 SPECweb,SPECjms2,SPECviewperf,SPECjEnterprise、SPEC HPC、SPEC OMP、SPEC MPI 等各种系列版本。由于 SPEC 组织的成员如 Intel、Oracle、IBM 等都是世界知名的计算机厂商,所以 SPEC 基准程序的测试结果获得普遍的认同。

TPC 基准程序　一套用于事务处理和数据库性能测试基准程序的标准规范,目前被广泛用于计算机系统和数据库性能评估。TPC 不给出基准程序的代码,而只给出基准程序的标准规范。TPC 先后推出过 11 套基准程序,目前正在使用的是 TPC-C、TPC-E、TPC-H。其中 TPC-C 测试模拟了一个比较复杂的在线事务处理应用环境,给出两个基准测试指标——性能指标 tpmC(每分钟处理的事物总数)和性价比指标 $/tpmC。随着 B2B、B2C 等新型应用逐渐兴起,TPC 组织推出了以美国纽约证券交易所为模型的 TPC-E,其测试结果的指标是 tpsE(每秒钟处理的事物总数)和 $/tpsE。TPC-H 是用来模拟决策支持类应用的一个测试集,其度量单位是每小时执行的查询数(QphH@size),size 表示数据库规模的大小。

参考文献

1. http://en. wikipedia. org/wiki/Benchmark_(computing)

2. http://www. netlib. org/benchmark/

3. Kant, Krishna. Introduction to computer system performance evaluation. New York: McGraw-Hill Inc. 1992

詹文岛

jishi tongxin

即时通信(instant message, IM)　允许两人或多人通过网络即时的传送文字信息、档案、语音和视频进行交流的通信方式。即时是指用户可以立刻与联络人进行交流,这通过即时通信所提供的状态信息特性来保证。大部分即时通信提供了状态信息的特性,用来显示联络人名单,联络人是否在线上以及能否与联络人交谈等。

即时通信最初只有发送即时文本信息等简单功能,此后陆续又具有文件传输、音视频聊天及网络游戏等更高级的功能。

最早的基于网络的即时通信形式可追溯到 20 世纪 70 年代的柏拉图系统(PLATO system)。之后

在 1980 年,UNIX/Linux 的即时交谈功能被广泛地使用于工程与学术界。20 世纪 90 年代,即时通信已成为跨越网间的交流方式。1996 年 11 月,ICQ 是首个广泛被非 UNIX/Linux 使用者用于互联网的即时通信软件。

随着互联网技术的迅速发展,即时通信也在不断地发生变化。自 1996 年第一个即时通信产品 ICQ 发明后,即时通信的技术和功能也开始基本成型,语音、视频、文件共享、短信发送等高级信息交换功能都可以在即时通信工具上实现,功能强大的即时通信软件足以搭建一个完整的通信交流平台。

当前使用的即时通信系统大都组合使用了 C/S 和 P2P 模式,在登录 IM 进行身份认证阶段是工作在 C/S 方式的。在这个过程中,用户首先连接服务器通知其在线状态,服务器收到此信息后,根据用户存储在服务器上的联络人列表将用户在线的信息发送给在线的联络人。之后服务器将用户存储在服务器上的联络人列表及相关信息回送到用户本地。当用户与联络人进行通信时,一般采用 P2P 模式。根据联络人列表中的信息,用户可以知道其联络人的 IP 地址及侦听端口,从而可以直接建立 TCP 或 UDP 连接进行通信。

即时通信在商业上的成功促使大量即时通信系统的出现。而由于服务提供商为了自身利益的考虑,使得不同的即时通信系统间难以相互交流。针对这种情况,IETF (Internet Engineering Task Force) 开始了即时通信的标准化工作。

IETF 把 IM 划分为四种协议,即 IMPP (即时信息和出席协议)、PRIM(出席和即时信息协议)、SIMPLE(针对即时信息和出席扩展的会话发起协议)及 XMPP (可扩展的消息出席协议)。PRIM 与 XMPP 类似,已经不再使用。比较有影响的是 SIMPLE 和 XMPP。

即时通信所面临的问题主要在于:①即时通信软件之间的互联通问题。②安全问题。不论个人用户,或是企业用户都面临着安全问题。由于即时通信的特点是两台终端之间直接进行交流,不必通过第三方服务器中转,这样也大大增加了即时通信用户的数据交换的监控的难度。

参考文献

1. 朱和平. 即时通信研究综述. 现代计算机, 2006

2. 佘其炯. 即时通信的现状与发展趋势. 北京: 电子工业出版社, 2007

3. 王海涛, 等. 即时通信——原理、技术和应用. 信息通信技术, 2010

(陈贵海)

jicheng dianlu zhizao

集成电路制造(integrated circuit manufacturing) 按照电路功能的设计要求,采用特殊的数据微缩制版、薄膜淀积和套版蚀刻等技术,将微电子设备中各种规格的元件(导线、电阻、电容和电感)和各种性能的器件(如二极管和晶体三极管)直接制造集成在一块半导体芯片上,组成一个不可分割的电路整体,完成其预定的电路功能。所谓集成是指在制造元器件的过程中同时完成对各元器件的连接,在连接元器件的过程中又同时整体地完成元器件的制造。

电子设备小型化的发展结果促使人们打破了传统的电路设计和制造概念。以往的电路设计都是将分立的元器件按要求组合和装配在一起,然后用导线加以焊接而成。1958 年美国人 J. Kilby 发明了集成电路,次年美国的得克萨斯仪器公司和仙童半导体公司分别实现了第一块集成电路。此后,集成电路经历了 20 世纪 60 年代初期的小规模(SSI)、60 年代中期的中规模(MSI)、70 年代初期的大规模(LSI)、70 年代中期的超大规模(VLSI)、90 年代的特大规模(ULSI)以及之后的巨大规模集成电路(GSI)阶段。高密度、低功耗、高可靠性和低价格是集成电路的特点(参见**计算机芯片**)。

集成电路制造的组织体系 现代集成电路制造的基础是硅平面扩散工艺,即用二氧化硅作为杂质扩散阻挡层(见图 3(f))的方法来制造晶体管。采用硅单晶作为集成电路的基础材料的原因是它具有合适的禁带宽度(1.12 eV)、稳定的氧化物以及在自然界中丰富的储藏量。除硅之外还采用锗、锗硅以及砷化镓、磷化铟等Ⅲ-Ⅴ族化合物半导体基础材料。

现代集成电路的制造工艺一般都应在超净化工作室内进行。室内的空气保持在良好控制的温度和湿度下,并且能连续地循环和过滤。按空气中微粒净化级别的定义,在"0.5 μm 的 M2 级"的净化环境是指:空气中大小在大于、等于 0.5 μm 的颗粒每立方米不超过 100 个。在关键的光刻工艺区域,净化条件应保持在"0.3 μm 的 M1 级"以下,即空气中大于、等于 0.3 μm 的颗粒每立方米不超过 3.09 个。

集成电路制造需要大量的纯水以供硅片在工序前的清洗。普通水中的颗粒(矿物质,动、植物遗体等)以及钠、钙、镁、铜和铁等离子都会沉积在硅片

上，导致器件性能的退化和失效。因此，采用超纯度、电子级水平的去离子水和化学试剂是制造集成电路的基本条件之一。在室温下，制造集成电路所使用的去离子水的电阻率应高于或等于 $18 \times 10^6/(\Omega \cdot cm)$。

集成电路的制造一般需要经过电路设计，工艺模拟，版图设计，衬底单晶硅片的选用，工艺集成，中间监测，划片、装架、封装，老化筛选，成品率统计和可靠性鉴定等工序。制造集成电路的组织体系如图 1 所示。

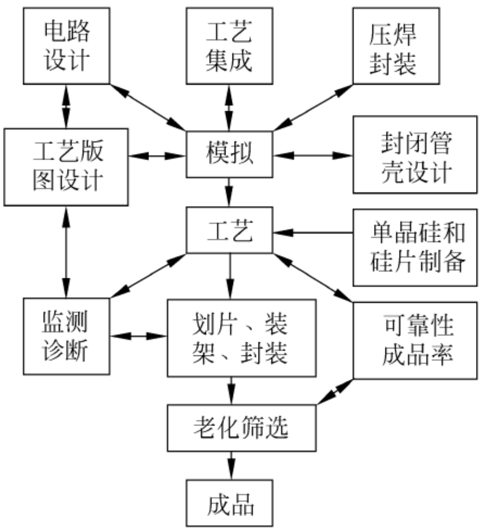


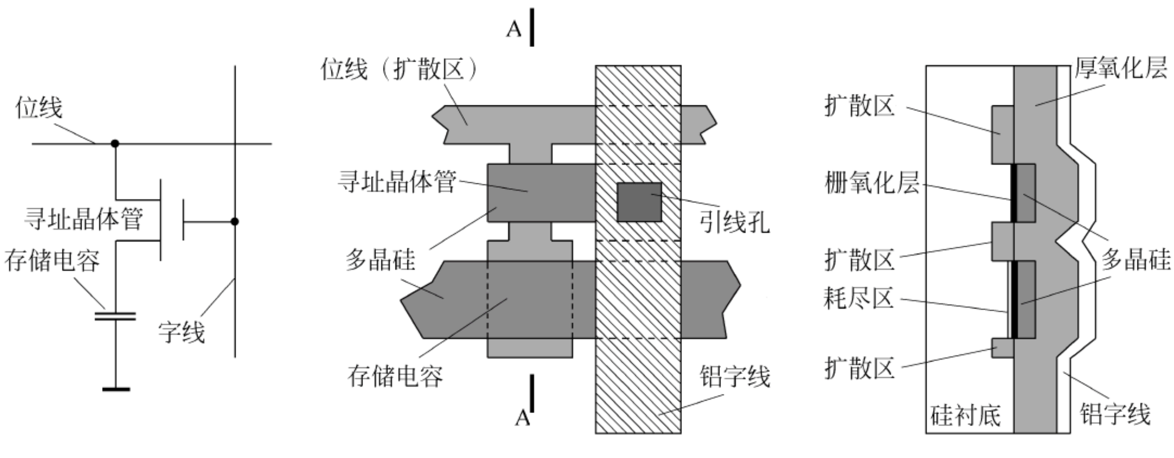
图 1　制造集成电路的组织体系

集成注入逻辑（I^2L）器件因其超高速逻辑运算能力，N 沟道金属-氧化物-半导体（NMOS）器件因其高密度、低功耗特性而被广泛用于计算机工业中。在各种存储器芯片中，以随机存取存储器（RAM）芯片的元器件密度最高。这是因为在 RAM 芯片中，矩阵中任何一位的信息都能独立存取。每一列存储位可利用扩散区、掺杂多晶硅线条或金属薄膜层的导电字线来实现存取。类似矩阵的列、行则可用图中位线来存取。动态 RAM 为保持存储的信息，要求存储在每个存储单元中的电荷数据周期性地“再生”。图 2 所示的是一种基本的单管动态存储单元。其中图 2(a) 为电路图，图 2(b) 为由 4 块光刻掩膜版（扩散区版、多晶硅版、孔版和铝引线版）组成的单元版图，图 2(c) 为图 2(b) 的 A—A 剖面。由扩散区（源和漏区）形成位线，扩散区即为晶体管的源电极。通过字线将该寻址晶体管的栅电极选通，存储信号（1 或 0）通过位线被读写。

现代集成电路制造是个极复杂、极精密的过程。尽管一套给定的流程包含的工艺步骤可以多达百余道，但是，只要工艺控制严格，重复性好，成品率可以较高；加上每一批流程中有近百片的硅片，每一片硅片上又有几百、几千粒 VLSI 的管芯（产品）同时在一起加工，因此每个产品的成本还是相当低的。

集成电路制造工艺　在集成电路制造中，电路设计、工艺模拟和版图设计是制造前的准备阶段。它们不仅关系到最终完成的集成电路能否按设计要求完成指定的功能，还关系到产品成品率的高低。集成电路制造工艺可以粗略地分成以下几个重要过程。

（1）单晶硅和硅片的制备　单晶硅和硅片的制备是集成电路制造前期工序之一。首先用化学方法把自然界的砂子（SiO_2）制成多晶结构的电子级硅棒，再根据晶体生长所涉及的从固体、液相或气相到结晶固相的相变这一物理过程，将多晶硅棒拉制成单晶硅棒，最后经切割、磨边、双面研磨、腐蚀、抛光



(a) 电路图　(b) 存储单元版图（由4块版图组成）　(c) A—A剖面图

图 2　单管动态存储单元

等工序将单晶硅棒制成单晶硅片。单晶硅片具有一定的晶向和导电类型,而电阻率、缺陷和位错密度、含氧量等参数都需要满足一定的设计要求。

(2) 工艺集成　工艺集成是集成电路制造的核心。图 3 为集成电路制造的部分基本集成工艺。其中图 3(a)为衬底单晶硅片,图 3(b)表示在硅片上用高温热生长法生长一层氧化硅(SiO_2)后再涂敷上一层抗蚀剂,例如正性胶。图 3(c)、(d)和(e)为光刻工序,目的是将设计的掩膜版上的图形精确无误地转移到 SiO_2层上。详细过程如下:利用照相曝光技术,把掩膜版上图形复制在抗蚀剂上,经显影后,掩膜版上的图样空白部分所对应的抗蚀剂(也称光

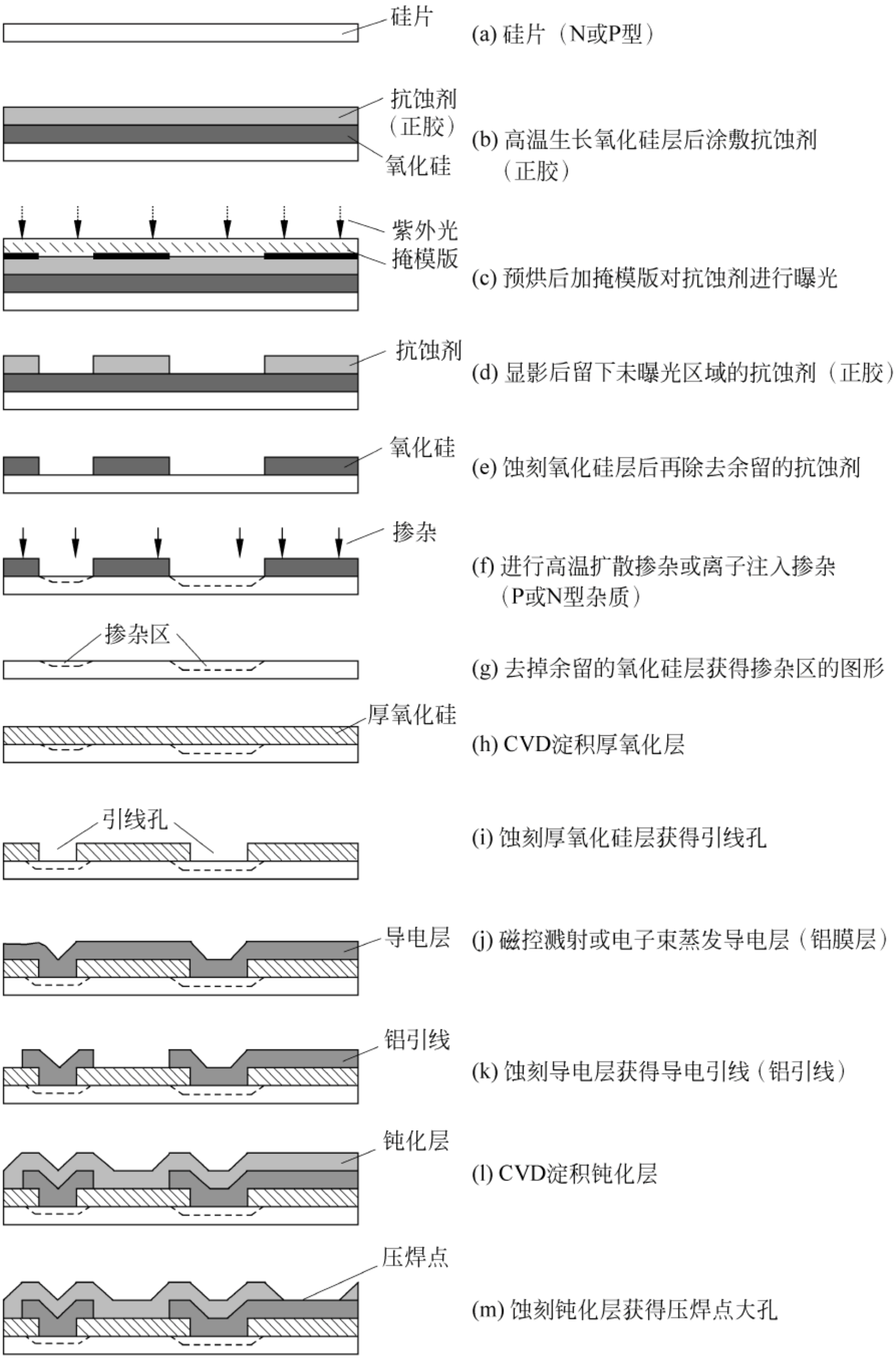


图 3　集成电路制造的部分典型集成工艺

刻胶)被溶解掉(曝光后可溶解的光刻胶称为正胶,曝光后不可溶解的光刻胶称为负胶)。留下的光刻胶经过烘烤、坚膜以后便具有抗蚀性能。经过等离子蚀刻把裸露的 SiO_2 层蚀刻掉,在抗蚀剂下面的 SiO_2 被保护而留下。再除去 SiO_2 上的抗蚀剂,最终使掩膜版上的图形完整无缺地、精确地复制在氧化硅层上,如图3(e)所示。图3(f)为掺杂剂通过高温热扩散或离子注入的方法进入硅表层的过程。掺杂剂可以是硼、磷、砷等离子或原子。氧化硅层起着掩蔽掺杂的作用,若采用离子注入,氧化硅层上的抗蚀剂常被保留,它同样具有阻挡掺杂离子或原子进入硅中的作用。被掺杂的区域叫掺杂区或扩散区。掺杂区在双极型器件中常用来形成基极、发射极和电阻;在MOS器件中则常用来形成源区、漏区、N阱和P阱,还用作对夹断电压大小的控制等。图3(g)表示蚀刻掉 SiO_2 层,留下所需要的扩散区域。在实际集成电路制造过程中,常常多次执行图3(b)到图3(g)的工序,只是用不同的版图以完成多块掩膜版的嵌套。图3(h)为在衬底硅片上淀积上一层起绝缘和隔离作用的厚氧化层。图3(i)为刻孔工序,在需要连接的各掺杂区上方将厚氧化层开个孔,形成带引线孔的氧化层。图3(j)为金属化过程,即在整个衬底硅表面溅射(或蒸发)上一层金属导电膜,金属膜通常为铝层。图3(k)表示按照实际线路的要求,光刻、刻蚀铝层,使硅片浅表层的各元器件按设计要求而相互连接。图3(l)为淀积钝化层。钝化层起着与外界的空气、水汽隔绝,保护管芯免受侵蚀和碰伤、划伤的作用。图3(m)为蚀刻压焊孔。按光刻、蚀刻的相同方法,在钝化层上蚀刻出较大的压焊孔,以便管芯在划片、装架后通过热压焊式超声压焊方法,用直径20~30 μm的铝丝将芯片各压焊点与对应的管座引线端连接。经过由图3(a)到图3(m)最基本的工艺集成,一个集成电路的芯片被制造出来。

(3) 管芯装配和老化筛选　管芯(也叫芯片)与管座的装配、老化筛选等都是集成电路制造的后工序。装配包括划片、装架、封装等工序,装架又分芯片的嵌装、压焊等工序,封装包括注入填料、焊接或粘接盖板等工序。封装后再进行老化筛选,对一个个集成电路成品进行“磨合”,完成严格的性能检测和分类。只要在一定的规格和等级允许的条件下使用,就可确保每一个集成电路产品都有较高的稳定性和可靠性。

在集成电路制造的各个工序环节,几乎都有相应的工艺质量监测和分析诊断技术。严格的质量管理和控制措施是保证集成电路产品的高质量、高可靠性以及高成品率的必要条件。

集成电路制造的关键因素　集成电路的迅速发展在很大程度上决定于设备和技术的发展。一代的设备,有一代的工艺;一代的技术,有一代的产品。与日俱进的超精细加工、薄膜淀积和离子注入的设备和技术是集成电路迅速发展的三大关键因素。

(1) 超精细加工　在超精细加工中,第一步是将精细的图形从掩膜版转移到硅表面上,图形转移的质量取决于光刻机和抗蚀剂的性能及其使用方法。由接触式光刻机、接近式光刻机、扫描投影光刻机、分步重复光刻机到步进扫描光刻机,已历经了五代光刻机的发展,由精细到微米级的紫外光光学光刻技术已发展成精细到微纳米级的电子束光刻和X射线光刻技术。超精细加工的第二步是对精细图形的蚀刻。由于湿法蚀刻技术的均匀性和可靠性较差,因此在超精细加工中逐渐被淘汰。干法蚀刻发展甚快,干法蚀刻有等离子蚀刻(IE)、反应离子蚀刻(RIE)、离子铣(IBE)和感应耦合等离子(ICP)蚀刻等形式,其中最常用的是IE和RIE两种。这些形式的主要差别在于蚀刻的方向性、选择性及蚀刻速率。

(2) 薄膜淀积　集成电路中的薄膜淀积方法常用真空蒸发、磁控溅射、化学气相淀积(CVD)和电镀淀积薄膜。利用光照或等离子体的作用可把CVD的淀积温度降下来。由于低压化学气相淀积(LPCVD)具有稳定可靠,重复性、均匀性好,适合大批量生产等优点,因此被广泛使用。在真空蒸发基础上发展起来的分子束外延(MBE)以及在CVD基础上发展起来的金属有机物化学气相淀积(MOCVD)都用来生长极薄的外延层。

(3) 离子注入技术　离子注入是一种向硅衬底中引入可控制数量和可控制深度的杂质,以改变其电学性能的方法。离子注入能够均匀、重复控制注入杂质的浓度和深度,而且可在低于125℃下进行操作,因而在所有应用中都优于高温扩散,已成为超大规模集成电路制造中最先进的技术。不同的工艺对离子注入的要求也各不相同。在MOS器件中利用离子注入机可完成深埋层、倒掺杂阱、穿通阻挡层、阈值电压调整、轻掺杂漏区(LDD)、源漏注入、多晶硅栅、沟槽电容器、超浅结、绝缘体上硅(SOI)等各种离子注入。离子注入后需要经过退火把注入的杂质元素激活并消除因注入而引起的晶格损伤。

为防止注入杂质在退火过程中向衬底内部再扩散，已越来越多地采用快速退火技术(RTA)。

展望 集成电路所用的主要衬底材料是单晶硅，但单晶砷化镓材料在集成电路中所占地位将越来越重要。各种超晶格材料不仅会改善某些现有的集成电路器件的性能，而且将产生一些有新效应的新器件。

新结构 新结构的集成电路发展趋势主要有以下几方面：①在绝缘衬底上生长单晶硅薄膜(SOI)技术。此技术最诱人的用途是制造三维集成电路，由二维发展到三维可以大大提高空间集成度。②多层布线技术。多层布线可以解决单层布线中常遇到的交叉线困境，实现多层布线的关键是解决中间绝缘层的表面平坦化问题。③耐熔金属硅化物和浅结构技术。耐熔金属硅化物与通常的掺杂多晶硅相比具有较低的自由电阻，此外，它还具有与硅接触点的长期稳定和可靠性，不易发生“穿刺”现象。④深槽结构技术。在VLSI和ULSI结构中，深槽的用途有两个方面：一是用作隔离，二是用来制作垂直方向的电容器，使电容器所占面积进一步缩小，从而增加器件的集成度。⑤量子阱、异质结等结构将使半导体光电集成电路的性能不断地提高。

新工艺、新技术 ①为减少互连线的电阻率，降低功耗，增加运算速度，铜质互连线必将取代传统的铝质互连线。②为制造多层结构的三维高密度集成电路，必须使用先进的化学机械抛光(CMP)平坦化技术代替传统的硼磷硅玻璃回流和旋涂膜层的平坦化方法。③倒装芯片技术是将芯片的有源面(具有表面键合压点)面向基座的粘贴封装技术。它是从芯片器件到基座之间最短路径的一种封装设计，为高速信号提供了最良好的电接触。由于不使用引线框架或塑料管壳，因此重量和外形尺寸也有所减小。④为获得更低成本、更加可靠、更加快捷、更高密度的电路，必须采用先进的集成电路封装技术，它们是球栅阵列(BGA)、板上芯片(COB)、表面安装技术(SMT)、多芯片模块(MCM)、芯片尺寸封装(CSP)及圆片级封装(WLP)等。

参考文献

1. Gise P E, Blanchard R. Semiconductor and integrated circuit fabrication techniques. Virginia: Fairchild Corporation, 1979

2. Michael Quirk, Julian Serda. Semiconductor manufacturing technology. New York: Prentice Hall, 2001

(汪锁发)

jichenghua nengli chengshudu moxing

集成化能力成熟度模型(capability maturity model integration for development, CMMI)

针对开发的一个有关产品和服务的过程改善的成熟度模型。其中过程改善是指人为设计的一个活动程序，其目的是改进组织的过程性能和成熟度，并改进这一程序的结果。

CMMI基于的基本思想是，支撑软件系统/产品质量有三大要素，分别为：①有素质的开发人员；②规程和方法；③工具和设备。通过软件过程可集成这三大要素，即过程可以有效地绑定开发人员、规程和方法以及工具和设备，以期按预期的进度和成本生产出高质量的软件系统/产品，即“有好的过程就有好的产品质量”。这一思想还可有效地应对软件开发环境的变化，包括软件技术不断发展，开发人员相对频繁流动。

CMMI组合了三个模型：软件能力成熟度模型(SW-CMM) V2.0 [SEI 1997]、系统工程能力模型(SECM)[EIA 1998]以及集成产品开发能力成熟度模型(IPD-CMM)[SEI 1997]，并在CMMI的基础上开发了针对“服务”的CMMI(v1.2 2007)和针对“获取”的CMMI(v1.2 2007)。

在CMMI中，规约了22个过程域，分别为：项目规划、项目监控、定量项目管理、集成项目管理、风险管理、提供方协定管理、需求开发、需求管理、技术解决方案、产品集成、确认、验证、配置管理、过程和产品质量保证、测量与分析、原因分析与解决、决策分析与解决、组织过程定义、组织过程性能、组织过程培训、组织过程关注、组织创新与部署。这些过程域覆盖了从概念到交付和维护的整个软件产品/系统生存周期。CMMI通过专用目标给出每个过程域要满足的独有特征以及实现诸专用目标的专用实践；通过共用目标给出诸过程达到一个特定制度化程度而必须呈现的特征以及相关的共用实践。

过程改善关键取决于实现以上各过程功能的制度化程度，为此CMMI规约了以下适用于所有过程域的五个共用目标：

(1)共用目标1(GG1) 达到专用目标，即该过程通过将认同的输入工作产品转换为认同的输出工作产品，支持该过程域并能使之达到其专用目标。

(2)共用目标2(GG2) 把该过程制度化为一个已管理过程，即制度化该过程，使之成为一个已管理过程。

(3)共用目标3(GG3) 制度化一个已定义的

过程,即制度化一个过程,使之成为一个已定义的过程。

(4) 共用目标 4(GG4) 制度化一个已定量管理过程,即制度化一个过程,使之成为一个已定量管理过程。

(5) 共用目标 5(GG5) 制度化一个已定量管理过程,即制度化一个过程,使之成为一个持续改进过程。

为了支持组织的过程改善,CMMI 提供了两种过程改善途径,一种是能力等级,该路径可使组织针对单一过程域,不断改善之;另一种是成熟度等级,该路径可使组织通过关注一组过程域,不断改善它们,以提高整个组织的过程性能。

能力等级是针对一个特定过程域的改善,分为六个等级,分别为:0 级—未完成级(incomplete)、1 级—已执行级(performed)、2 级—已管理级(managed)、3 级—已定义级(defined)、4 级—已定量管理级(quantitatively managed)、5 级—持续优化级(optimizing)。在 CMM1 中,为一个特定过程域的改善,就每一能力等级规定了相应的专用目标以及实现专用目标的专用实践—实现该过程功能的最佳实践;规定了一个共用目标以及实现该共用目标的共用实践,即 1 级到 5 级分别要达到共用目标 1 到共用目标 5。

成熟度等级是针对预先定义的一组过程域的改善,分为 5 级,分别为 1 级—初始级;2 级—已管理级;3 级—已定义级;4 级—已定量管理级;5 级—持续优化级。除了初始级之外,2 级包含七个过程领域,分别为配置管理、测量与分析、项目监控、项目规划、过程和产品质量保证、需求管理、提供方协定管理;3 级包含十一个过程领域,分别为决策分析与解决、集成项目管理、组织过程定义、组织过程关注、组织培训、产品集成、需求开发、风险管理、技术解决方案、验证、确认;4 级包含二个过程领域,分别为组织过程性能和定量项目管理;5 级包含二个过程领域,分别为原因分析与解决、组织创新与部署。在 CMMI 中,为每一成熟度等级中所包含过程域的改善,规定了相应的专用目标以及实现专用目标的专用实践;规定了相应的共用目标以及实现共用目标的共用实践,即成熟度 1 级要达到共用目标 1;成熟度 2 级要在共用目标 1 的基础上达到共用目标 2;成熟度 3 级要在共用目标 2 的基础上达到共用目标 3;4 级和 5 级至少要达到共用目标 3。

总之,CMMI 包含的最佳实践,包括专用实践和共用实践,覆盖了产品从概念到交付和维护的整个生存周期,强调了构造和维护当今产品所必要的工作,并给出过程改善的两种途径。系统/产品开发组织可根据实际业务工作需要以及资源情况,灵活地进行过程改善,不断提高系统/产品的质量。

(王立福)

jicheng xuexi

集成学习(ensemble learning) 利用多个学习器解决问题的一种**机器学习**范式。狭义地说,集成学习利用多个同质(homogeneous)学习器对同一个问题进行学习。“同质”是指所使用的学习器属于同种类型。广义地说,只要是使用多学习器解决问题,就属于集成学习。在不同场合下,集成学习也被称为多分类器系统(multi-classifier system)、基于委员会的学习(committee-based learning)等。

利用多学习器解决问题是人类进行问题求解的一种基本想法,已很难考证其最早出现的时间。一般认为,集成学习大致在 20 世纪 90 年代初开始成为机器学习的一个研究领域。

集成由若干个学习器组成。在同质情形下,这些学习器是由一个基学习算法(base learning algorithm)产生,因此被称为基学习器(base learner),常用的基学习算法包括**决策树**、**人工神经网络**等;在异质情形下,集成中包含不同类型的学习器。无论何种情形,集成中的学习器都可称为个体学习器(individual learner)或成分学习器(component learner)。通过将学习器进行有效的结合,集成的泛化(generalization)能力通常优于单个学习器。一般而言,有效的集成学习方法可以将泛化能力仅比随机猜测略好的弱学习器(weak learner)提升为泛化能力渐近理想的强学习器(strong learner),因此,集成中的学习器有时也被直接称为弱学习器。

构造集成的过程通常包含两步:首先从训练样本集中产生若干个体学习器,然后将这些个体学习器进行结合。根据个体学习器产生的方式,常见的集成学习方法可大致划分为“并行式方法”和“串行式方法”。在并行式方法中,不同的个体学习器可同时产生,基于自助采样(bootstrap samping)的 Bagging 算法是此类的代表;在串行式方法中,先产生的个体学习器将对后产生的个体学习器发生影响,基于统计残差逼近的 Boosting 算法是此类的代表。个体学习器的结合方式多种多样,最常见的有投票法、平均法等;还可通过学习器进行结合,例如 stacking 方法。

一般认为,个体学习器之间的差异(diversity)对集成的泛化能力有重要影响,个体学习器泛化能力越好,个体之间差异越大,则集成的泛化能力越强。因此,在获得个体学习器之后,选择泛化能力较强且差异较大的个体(尤其是"互补性"强的个体)构建集成,往往比使用全部个体学习器构建集成更好。从**统计学习**的角度来看,这样的"选择性集成"过程在一定程度上起到了类似于 L1 正则化的作用,通过对模型的复杂度进行控制而获得更强的泛化能力。

参考文献

1. Zhou Z H. Introduction to ensemble methods. New York: Taylor and Francis/CRC Press, 2012

2. Rokach L. Pattern classification using ensemble methods. Singapore: World Scientific Press, 2010

3. Kuncheva L I. Combining pattern classifiers: Methods and algorithms. Hoboken, NJ: JohnWiley & Sons Press, 2004

(周志华)

jihe

集合(set) 由一定范围的、确定的且彼此可以区别的对象(抽象的或具体的)组成的整体。它是数学中最重要的基本概念之一。组成集合的对象称为它的元素。例如"中国人的集合"、"长江里的鱼的集合"及"方程 $x^3-4x+3=0$ 的实根的集合"等,都是集合的例子。

属于关系 若对象 a 是集合 A 的元素,则称 a 属于 A 或 A 含 a,记为 $a\in A$;否则称 a 不属于 A 或 A 不含 a,记为 $a\notin A$。不含任何元素的集合称为**空集**,用∅表示。

子集 若集合 A 的元素都是集合 B 的元素,则称 A 是 B 的子集或 B 包含 A,记为 $A\subseteq B$。

相等 若集合 A,B 的元素完全相同,则称 A 与 B 相等,记为 $A=B$;否则称 A 与 B 不相等,记为 $A\neq B$,显然,$A=B$ 当且仅当 $A\subseteq B$ 且 $B\subseteq A$。

若 $A\subseteq B$ 且 $A\neq B$,则称 A 为 B 的**真子集**,记为 $A\subset B$。

对任意元素 a 和集合 A,恒有 $a\notin\varnothing$,$\varnothing\subseteq A$ 且 $A\subseteq A$。

集合的常用表示方法有以下三种:

(1)列举法 依照任意次序不重复地列举出集合的全部元素,并用花括号括起来。例如,

10 以内全体素数的集合 $=\{2,3,5,7\}$。

方程 $x^4-4x+3=0$ 的全部实根的集合 $=\{-1,1,2\}$。

(2)命题法 若集合 A 的全体元素恰为使命题 $P(x)$ 真的全体 x,则 A 可用 $\{x\mid P(x)\}$ 表示,即 $A=\{x\mid P(x)\}$。例如,有理数集合

$$\mathbf{Q}=\left\{\frac{m}{n}\;\middle|\;m\text{ 和 }n\text{ 为整数且 }n\neq 0\right\}$$

$$\{1,2\}=\{x\mid x\text{ 为方程 }x^2-3x+2=0\text{ 的实根}\}$$

(3)归纳定义法 用这种方法定义一个集合 A 时,一般包括以下三个步骤:

① 基本项:已知非空集合 $S_0\subseteq A$。② 归纳项:给出一组规则,从 A 的任意元素(一个或多个)出发,依据这组规则所得之元素都是 A 的元素。③ A 中每个元素都可以通过有限次应用①和②得到。

步骤①是归纳定义法的基础,且保证 $A\neq\varnothing$;步骤②是归纳定义法的关键步骤,用以从 A 的已知元素获得其新元素;步骤③保证所定义的集合 A"最小",因而是唯一的。在用归纳定义法给出集合时,步骤③常常省略不写。

例如,全体奇数的集合 O 可用归纳定义法给出如:①$1\in O$;②若 $n\in O$,则 $n+2\in O$;③省略。

(王兵山)

jihelun

集合论(set theory) 以一般集合为研究对象的数学的基本分支之一。集合论在数学中占有独特的地位,它的基本概念已渗透到数学的所有领域。按现代数学的观点,数学各分支的研究对象,或者是带有某种结构的集合(如群、环、域、拓扑空间等),或者是可用集合直接定义(如自然数、有理数、实数、函数等),或者是可借助集合定义(如范畴、函子、自然变换等)。因此,从某种意义上说,集合论是整个现代数学的基础。

集合论是 G. Cantor 于 19 世纪末创立的,至今经历两个阶段:1908 年以前称朴素集合论,1908 年以后又产生了公理集合论。公理集合论不外乎是朴素集合论的严格处理,由于广泛使用数理逻辑的工具,它又逐渐成为数理逻辑的一个重要分支。自 20 世纪 60 年代以来,它又获得迅速发展。

朴素集合论 它是 G. Cantor 最早建立起来的集合论,故又称康托尔集合论。所谓一个集合,按 G. Cantor 的定义,就是一定范围的、确定的且彼此可区别的对象(抽象的或具体的)汇集在一起组成的一个整体。组成集合的对象称为它的元素或成员。如果对象 x 是集合 A 的元素,则记为 $x\in A$,否则记为 $x\notin A$。因为在 G. Cantor 的集合定义中,对于对象

的所属范围、性质及其如何汇集都没做要求，所以它还构不成集合的严格数学定义，而是集合的一种直观描述或说明。

在朴素集合论里，无条件地接纳了以下三条基本原理：

(1) 外延原理　两个集合相等是指它们的元素完全相同；

(2) 概括原理　对任意性质 P，都有一个使 $P(x)$ 真的全体 x 的集合，记为 $\{x|P(x)\}$；

(3) 选择公理　对每个由非空集合组成的集合族 $\mathscr{S}$，都有一个函数 f（称为 $\mathscr{S}$ 的选择函数），使得对任意 $A \in \mathscr{S}$ 皆有 $f(A) \in A$。

选择公理还有其他多种表达形式。这条公理的直观意思是说，可从任何非空集合内取得一个元素。选择公理对整个数学的影响是巨大的，它在数学的许多分支，如分析学、拓扑学、代数学等中都是不可少的工具。

有了集合概念，就可以定义一个集合 A 的子集 $S \subseteq A$，幂集 $\mathscr{P}(A)$ 和补集 $\sim A$，两个集合 A 与 B 的并集 $A \cup B$、交集 $A \cap B$、差集 $A - B$ 和笛卡儿积 $A \times B$ 等，并可进而定义集合上的关系、集合到集合的函数及集合的基数（也称势）和序数等一系列概念。关于它们的运算和性质的研究，就构成了朴素集合论的主要内容。

集合论悖论　悖论就是逻辑矛盾的论述。在朴素集合论内，依据概括原理，对任意性质 P，都有一个使 $P(x)$ 真的全体 x 的集合 $\{x|P(x)\}$。如果取 $P(x)$ 为"x 是集合且 $x \notin x$"，便可获得一个集合 $S = \{x|x$ 是集合且 $x \notin x\}$。现在要问：S 是不是它自己的元素呢？若 S 是它自己的元素即 $S \in S$，则由 S 的定义及 S 是集合得 $S \notin S$；若 S 不是它自己的元素，则由 S 的定义及 S 是集合得 $S \in S$，总之恒有 $S \in S$ 当且仅当 $S \notin S$。这显然是一个矛盾，它就是著名的罗素悖论。此外，人们在朴素集合论内还相继发现了许多悖论。这类悖论表明，朴素集合论的理论是不协调的，这也使人们对整个数学推理的正确性与结论的真理性产生怀疑，酿成了数学史上的第三次危机，影响是深远的。

公理集合论　为了避免悖论，人们感到再不能把任何逻辑上可定义的概念的外延都当作一个集合了，即须对 G. Cantor 的集合定义加以限制。但一直未能找到一个简单而无问题的替代定义，于是就从当时已有的集合论成果出发，来反求足以建立这一数学分支的原则，这就是集合论的公理化研究。E. Zermelo 于 1908 年提出了第一个集合论公理系统，后经 A. A. Fraenkel 和 A. T. Skolem 加以改进和补充，形成了著名的 ZF 公理系统。罗素也于 1908 年提出了类型论，它是关于集合的型的层次理论。一个集合 x 能够是一个集合 y 的元素当且仅当 y 的层次恰比 x 的层次多 1，因此再不能讲"所有集合的集合……"。类型论包括简单类型论和分支类型论两部分。J. von Neumann 于 1925 年给出了把"类"作为合法数学对象的类公理。经过 P. Bernays 和 K. Gödel 的改进与完善，又形成了另一个著名的集合论公理系统，即 GB 公理系统。这些公理系统都能够描述朴素集合论的丰富内容，建立起朴素集合论的已有定理，排除朴素集合论中出现的悖论，并为解决集合论未解决的问题，特别是连续系统假设问题提供方便。

现在，公理集合论多采用形式化方法，利用数理逻辑的一阶谓词演算系统，建立起一阶集合论形式语言，把公理集合论的公理、定理和命题表示为该形式语言的合式公式（简称公式），从而便于用数学方法进行严格的推演与论证。因此，公理集合论不外乎是朴素集合论的严格处理。下面介绍最引人注目的 ZF 公理系统，有时为了强调选择公理，又把它称为 ZFC 公理系统。这个公理系统包括以下几条公理：

(1) **外延公理**

$$\forall A \forall B(\forall x(x \in A \leftrightarrow x \in B) \rightarrow A = B)$$

两集合 A 与 B 相等是指它们的元素完全相同，并记为 $A = B$。把 $\forall x(x \in A \rightarrow x \in B)$ 简写为 $A \subseteq B$，并称 A 为 B 的子集。

(2) **空集公理**

$$\exists x \forall y(y \notin x)$$

存在一个不含任何元素的集合，称为空集。根据外延公理，空集是唯一的，并用 $\varnothing$ 表示。

(3) **无序对公理**

$$\forall A \forall B \exists C \forall x(x \in C \leftrightarrow x = A \lor x = B)$$

对任二集合 A, B，都有一个恰以 A, B 为元素的集合，称为 A 与 B 的无序对，并记为 $\{A, B\}$。记号 $\{A\}$ 表示 $\{A, A\}$，并把 A, B 的有序对 $\langle A, B\rangle$ 定义为集合 $\{\{A\}, \{A, B\}\}$。

(4) **并集公理**

$$\forall A \exists B \forall x(x \in B \leftrightarrow \exists X(x \in X \land X \in A))$$

对任意集合 A，都有一个恰以 A 的元素的元素为元素的集，称为 A 的并集，并记为 $\cup A$。当 $A = \{X_1, X_2, \cdots,$

$X_n\}$时，常把$\cup A$表示为$X_1\cup\cdots\cup X_n$或$\bigcup_{i=1}^{n} X_i$。

(5) **幂集公理**

$$\forall A\exists X\forall B(B\in X\leftrightarrow B\subseteq A)$$

对任意集合A，都有一个恰以A的子集为元素之集合，称为A的幂集，并记为$\mathscr{P}(A)$或2^A。

(6) **无穷公理**

$$\exists x(\varnothing\in x\wedge\forall y(y\in x\rightarrow y\cup\{y\}\in x))$$

存在含有无穷多个元素的集合。

(7) **分离公理**(又称子集公理)

$$\forall x\exists y\forall z(z\in y\leftrightarrow z\in x\wedge A(z))$$

其中$A(z)$为ZF公式。

这不是一条公理，而是一个公理模式。当所涉及的对象都是某个大集合的元素时，就可以使用概括原理而获得该大集合一个子集。若A为任意非空集合，则由分离公理知道，$\forall A\exists B\forall x(x\in B\leftrightarrow\forall y(y\in A\rightarrow x\in y))$必定义一个集合，称为$A$的交集，并记为$\cap A$。当$A=\{X_1,\cdots,X_n\}$且$n\geqslant1$时，常把$\cap A$表示为$X_1\cap\cdots\cap X_n$或$\bigcap_{i=1}^{n} X_i$。

(8) **替换公理**

$$\forall x\exists!yA(x,y)\rightarrow\forall S_1\exists S_2\forall u(u\in S_2\leftrightarrow \exists z(z\in S_1\wedge A(z,u)))$$

其中$\exists!yA(x,y)$为$\exists yA(x,y)\wedge\forall z(A(x,z)\rightarrow z=y)$的缩写。

这也不是一条公理，而是一条公理模式。设$A(x,y)$为任意公式，若对任意集合x都有唯一的集合y使$A(x,y)$成立，则对任意集合S_1都有集合S_2使$S_2=\{u\mid$有$z\in S_1$使$A(z,u)$真$\}$。即S_2为恰由S_1中各元素经$A(x,y)$对应的值组成的集合。

(9) **正则公理**

$$\forall x(x\neq\varnothing\rightarrow\exists y(y\in x\wedge x\cap y=\varnothing))$$

对任意非空集合x，皆有集合y使$y\in x$且$x\cap y=\varnothing$。

正则公理说明集合与其元素之间具有某种层次关系。

(10) **选择公理**

$$\forall x(x\neq\varnothing\rightarrow\exists f\forall y(y\in x\wedge y\neq\varnothing\rightarrow f(y)\in y))$$

对任意非空集合x，都有一个函数(称为选择函数)f，使得当$y\in x$且$y\neq\varnothing$时，有$f(y)\in y$。

这个公理系统并不是独立的，如分离公理就可由其他公理推出。但由于历史原因及其使用的方便性，还是把它列入了。

ZF公理集合论不是完备的，因为G. Peano算术公理都是ZF公理集合论的定理，故由G. Gödel第二不完备性定理知道，它一定是不完备的。

参考文献

1. Takeuti G, Zaring W M. Introduction to axiomatic set theory. 2nd ed. New York: Springer, 1982

2. Enderton H B. Elements of set theory. New York: Academic Press, 1977

(王兵山　张景文　吕义忠)

jihe yunsuan

集合运算(**operations of sets**)　从已知集合获得新集合的常用方法。

在讨论某类问题时，通常有一个含有所涉及的全部元素之固定集合，称为全集或空间，常用U表示，其他集合全是U的子集。假定A与B为集合。

并　A与B的并集为集合$\{x\mid x\in A$或$x\in B\}$，记为$A\cup B$。

交　A与B的交集为集合$\{x\mid x\in A$且$x\in B\}$，记为$A\cap B$。

差　A与B的差集为集合$\{x\mid x\in A$且$x\notin B\}$，记为$A-B$或$A\backslash B$。

补　A的补集为集合$U-A$，记为$\sim A$。

对称差　A与B的对称差集为集合$(A\cup B)-(A\cap B)$，记为$A\oplus B$。

如果$A\cap B=\varnothing$，则称A与B不相交。

上述5种集合运算，可用图1所示的文氏图直观地表示，图中阴影部分为运算结果。

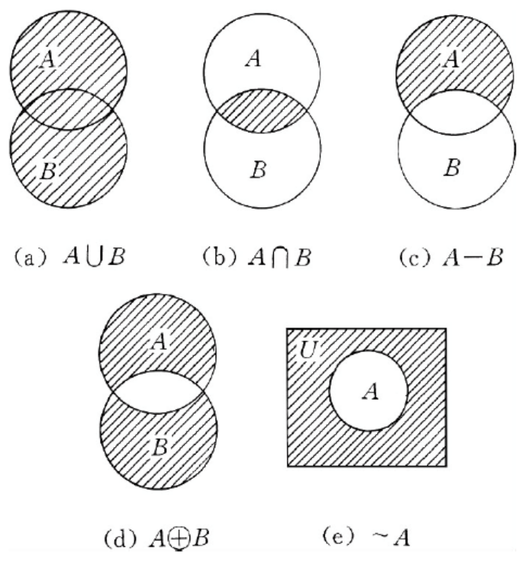


图1　集合运算的文氏图

例1　设$U=\{2,3,5,7,11,13\}$，$A=\{2,5,7\}$，

$B=\{2,3,7,11\}$，则

$$A\cup B=\{2,3,5,7,11\}$$
$$A\cap B=\{2,7\}$$
$$A-B=\{5\}$$
$$A\oplus B=\{3,5,11\}$$
$$\sim A=\{3,11,13\}$$

关于集合运算∪，∩和—，有以下基本定律：

幂等律

$$A\cup A=A$$
$$A\cap A=A$$

交换律

$$A\cup B=B\cup A$$
$$A\cap B=B\cap A$$

结合律

$$(A\cup B)\cup C=A\cup(B\cup C)$$
$$(A\cap B)\cap C=A\cap(B\cap C)$$

分配律

$$A\cup(B\cap C)=(A\cup B)\cap(A\cup C)$$
$$A\cap(B\cup C)=(A\cap B)\cup(A\cap C)$$

同一律

$$A\cup\varnothing=A$$
$$A\cap U=A$$

零律

$$A\cup U=U$$
$$A\cap\varnothing=\varnothing$$

互补律

$$A\cup\sim A=0$$
$$A\cap\sim A=\varnothing$$

吸收律

$$A\cup(A\cap B)=A$$
$$A\cap(A\cup B)=A$$

对偶律

$$\sim(A\cup B)=\sim A\cap\sim B$$
$$\sim(A\cap B)=\sim A\cup\sim B$$

对合律

$$\sim(\sim A)=A$$

关于集合的并和交两种运算，还可做如下推广。

广义并　A 的广义并为集合$\{x\mid$有集合 $S\in A$ 使 $x\in S\}$，记为$\cup A$。

当 $A=\{A_1,\cdots,A_n\}$ 且 $A_1,\cdots,A_n$ 均为集合时，则把$\cup A$ 记为 $A_1\cup A_2\cup\cdots\cup A_n$ 或 $\bigcup_{i=1}^{n}A_i$。

广义交　若 $A\neq\varnothing$，则 A 的广义交为集合$\{x\mid$若 $S\in A$，则 $x\in S\}$，记为$\cap A$。

当 $A=\{A_1,\cdots,A_n\}$ 且 $A_1,\cdots,A_n$ 均为集合时，则把$\cap A$ 记为 $A_1\cap A_2\cap\cdots\cap A_n$ 或 $\bigcap_{i=1}^{n}A_i$。

幂集　A 的幂集为集合$\{S\mid S\subseteq A\}$，记为 2^A 或 $\mathscr{P}(A)$。

这时显然有$\varnothing\in 2^A$ 且 $A\in 2^A$，而且当 $A\subseteq B$ 时还有 $2^A\subseteq 2^B$。如果 A 有 n 个成员，则 2^A 恰有 2^n 个成员。若 $A=\{a,b,c\}$，则

$$2^A=\{\varnothing,\{a\},\{b\},\{c\},\{a,b\},\{a,c\},\{b,c\},\{a,b,c\}\}$$

对任意两个元素，可把有序偶$\langle a,b\rangle$定义为集合$\{\{a\}\{a,b\}\}$，由此定义可知

$\langle a,b\rangle=\langle c,d\rangle$ 当且仅当 $a=c$ 且 $b=d$

笛卡儿积　A 与 B 的笛卡儿积是集合$\{\langle a,b\rangle\mid a\in A$ 且 $b\in B\}$，记为 $A\times B$。当 $A=B$ 时，又把 $A\times B$ 写作 A^2。

如果 $A=\{0,1\}$ 且 $B=\{a,b,c\}$，则

$$A^2=\{\langle 0,0\rangle,\langle 0,1\rangle,\langle 1,0\rangle,\langle 1,1\rangle\}$$
$$A\times B=\{\langle 0,a\rangle,\langle 0,b\rangle,\langle 0,c\rangle,\langle 1,a\rangle,\langle 1,b\rangle,\langle 1,c\rangle\}$$
$$B\times A=\{\langle a,0\rangle,\langle a,1\rangle,\langle b,0\rangle,\langle b,1\rangle,\langle c,0\rangle,\langle c,1\rangle\}$$

由此可知，$A\times B\neq B\times A$，即笛卡儿积不是可交换的。

集合 $A_1,\cdots,A_n$ 的笛卡儿积$(\cdots((A_1\times A_2)\times A_3)\cdots)\times A_n$简写为 $A_1\times A_2\times\cdots\times A_n$ 或 $\mathop{\times}_{i=1}^{n}A_i$，其元素$\langle\langle\cdots\langle\langle x_1,x_2\rangle,x_3\rangle,\cdots\rangle,x_n\rangle$简写为$\langle x_1,x_2,\cdots,x_n\rangle$，并称为 n 元序偶。

关于集合的笛卡儿乘积，有以下运算定律：

(1) $A\times B\subseteq C\times D$　当且仅当 $A\subseteq C$ 且 $B\subseteq D$

(2) $A\times B=C\times D$　当且仅当 $A=C$ 且 $B=D$

(3) $A\times(B\cup C)=(A\times B)\cup(A\times C)$
　$(A\cup B)\times C=(A\times C)\cup(B\times C)$

(4) $A\times(B\cap C)=(A\times B)\cap(A\times C)$
　$(A\cap B)\times C=(A\times C)\cap(B\times C)$

(5) $A\times(B-C)=(A\times B)-(A\times C)$
　$(A-B)\times C=(A\times C)-(B\times C)$

（王兵山）

jiqun jisuan

集群计算(cluster computing)　一种并行的计算方式。它将通用的高档微机或者**工作站**，通过系统级网络或者**局域网**连接起来，在操作系统与工具系统的支持下实现统一调度和协调处理。支持集群

计算的系统中的计算机(也称为集群计算机)可以是同构的,也可以是异构的,分别被称为是同构集群以及异构集群。如果所用的计算机是工作站,称为工作站机群或工作站网络(NOW)。工作站网络的主要思想起源于美国加州大学 Berkeley 分校,NOW 的思想是把校园中用网络连接起来的工作站的空余计算能力集中利用起来,用以完成一台工作站无法完成的工作。

集群计算机的特性

集群计算机具有用户投资风险小,编程方便,系统结构灵活,性能价格比高,可伸缩性好等特点。集群计算机使用了广泛被接受的服务器硬件以及通用网络,系统可大可小,适应性良好,能够降低用户的投资风险。采用的操作系统以及应用软件也是通常使用的操作系统以及相关的并行计算软件,在编程上方便用户的使用。

集群计算机除了高性能、低成本的优势之外,还有研制周期短的优点。集群计算机系统的建造主要是采用现成的与通用的或已商品化的软件和硬件,即所谓的 COTS(commercial off-the-shelf)方式。这样在大大降低研制成本的同时,也大幅度缩短了研制的周期。而且可以采用最新的软硬件技术以提高集群计算机系统的性能。

集群计算机设计和实现时要解决两个关键问题:一是从各个角度想方设法降低处理机或进程间的通信开销,使系统的资源主要用于计算,从而获得较高的加速比和利用率。采用的方法是针对并行计算的特点,研究和设计高效的精简通信机制和协议,并将底层协议用高速通信部件来支持,大幅度降低通信开销,提高并行效率,从而使系统的性能更高,适用面更广。同时采用快速以太网、光纤分布式数据接口或异步传送模式等技术。二是设计和实现一套友好的、高效的并行程序开发环境和工具系统。用户可以用系统支持的传统 FORTRAN,C,C++ 等语言编程,使用户能继承已有的软件财富。用户还可用系统提供的通信原语库、输出输入操作、并行图形处理、可视化性能评测工具、并行程序调试工具以及人机交互集成开发工具等并行处理工具包,对程序的并行运行进行控制、调试、监测和评价等工作。

高性能 CPU 和高速互联网络的普及是集群计算机出现的硬件基础。根据摩尔定理,中央处理器(CPU)的集成度大约每 18 个月翻一番,其计算性能也成倍地提高,而达到相同性能所需要的成本大幅度降低。这样,采用常见的通用 CPU 就能提供很高的计算能力。在通信技术方面,通信延迟迅速缩短,通信带宽成倍提高,在高效通信的基础上,就为多台计算机之间的并行计算提供了可能。在软件方面,大量的功能越来越强、性能越来越高的免费操作系统与通信软件的出现,使得多机并行变得很容易实现,不必从头开发相应的并行软件与系统。因此,集群计算机这种以相对独立的计算结点(独立的CPU,独立的内存,独立的操作系统)为单位的并行计算方式便得以成长并迅速普及开来。

集群计算机的软硬件环境

图 1 是集群计算机的硬件结构图。它由工作站和**互连网络**两部分组成,工作站上增加一个主机接口实现连网。工作站包含高速定点和浮点处理器、大容量存储器、良好的网络支撑环境和图形显示设备以及友好的用户界面。互连网络通常使用局域网,如**以太网**、权标环网、光纤分布式数据接口网等,也可以是交换网络,如异步传送模式(ATM)、交换式高速以太网等。

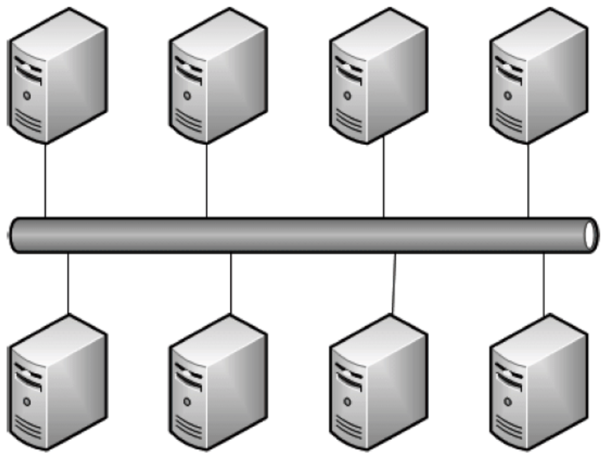


图1 集群计算机系统的硬件构造

图2是集群计算机的软件结构图。可以看到,整个软件系统都建立在通用的操作系统之上。为了支持某些特殊的操作,可以对原有的操作系统进行修改和重建。通信协议实现工作站到互连网络的数据通信服务。通信原语库由用户编程调用,并行程序设计环境和并行工具包为用户提供方便、友好的使用接口。

在并行编程上,则主要包括功能并行以及数据并行两个方面。在功能并行中,每台工作站并行执行应用程序的部分功能,而在数据并行中,每台工作站并行处理应用程序的部分数据。在两种并行模型中,互连网络都承担工作站间数据交换和信号传输的任务。规模较大的计算机集群系统采用的操作系统主要是Linux(参见**Linux 操作系统**),而并行程序设计环境主要是



图2　集群计算机中节点运行的软件情况

消息传递接口(MPI),**互连网络**主要是 Myrinet,Infiniband 和 Fast Ethernet。集群计算机的关键技术是减少消息传递的通信开销,建立并行程序设计环境以及开发集群管理系统与相关工具。

集群计算机的应用

高性能计算　几乎全球大部分的高性能计算机都是用集群技术构造的,主要的原因还是集群计算的快速构造以及可扩展特性。高性能集群计算的主要任务是将计算问题分配到集群的不同的计算节点中,并通过降低交互通信量,提高通信带宽与降低通信延迟,以达到快速计算的目的。高性能集群计算的计算能力非常高,对于浮点数的处理能力非常强,通常用于科学计算,数值模拟,金融计算等需要大规模超高速计算的领域。高性能集群计算机通常采用的是 Linux 操作系统以及其他的一些开源软件构成并行计算的服务,通常采用专门为科学计算所设计的 MPI 库来编写和运行并行程序。

大规模数据处理　近几年来,随着互连网络用户以及规模的不断扩大,网络应用对于大规模数据处理的需求越来越高,集群计算机也被大量应用于大规模数据处理。由于集群天然的可扩展性,存储规模也非常庞大,因此,许多网络服务都是使用集群计算机方式进行构造。在这些大规模数据处理的应用中,一般会构造一个分布式的存储系统例如分布式文件系统,将数据分散到集群中的各个节点。而后,通过一些编程模型方面的支持,在集群中对这些数据进行并行处理,并获得最终的结果。通过集群进行大规模数据处理的编程模式有 MapReduce 等,是当前流行的大规模数据处理方式。

参考文献

1. Sloan J D. High performance Linux clusters with OSCAR, Rocks, OpenMosix, and MPI. O'Reilly Media, 2004

2. 李建江,薛巍,张武生,等. 并行计算机及编程基础. 北京: 清华大学出版社,2011

（郑纬民）

jisan kongzhi xitong

集散控制系统(distributed control syetem, DCS)　利用计算机技术、信号处理技术、控制技术、网络通信技术和人机接口技术等对生产过程进行分散控制,集中监视、操作和信息管理的分布式计算机控制系统。它又称分散控制系统。

集散控制系统在体系上是一种分层分布的多处理机结构。它实现地理上和功能上的分散控制,同时通过高速通信网络(过程控制网络)把各个分散点的生产过程信息集中起来,进行集中监视、操作和信息管理,既克服了集中控制带来的危险性高度集中的问题,又避免了一般控制装置分散在各设备现场监视操作困难且对信息无法统一管理的弱点。它不仅提高了系统的可靠性,同时使系统扩展灵活,减轻了操作人员的劳动强度,减少了运行维护量,可实现连续控制、**顺序控制**、批量控制等不同的控制功能,甚至可具备实现预测控制、最优控制等先进控制的编程能力,因此适合现代工业大生产的要求。它能完全替代单元式常规控制仪表,在工业控制领域获得极其广泛的应用,是当前炼油、化工、冶金、制药、电力等流程化工业进行计算机实时控制和集中管理的主流设备。

集散控制系统的出现是工业控制历史上的一个里程碑。自 1975 年美国 Honeywell 公司成功地推出世界上第一套集散控制系统 TDC-2000 以来,它已经历了几十年的发展,产品已走向成熟,已经从初级的局部网络系统扩展成大型的分层网络系统。它的控制运算功能、应用规模、系统可靠性、信息处理能力等已有很大的提高,产品更加标准化、综合化、开放化和现场级的智能化,基于网络的控制系统规模更加灵活可变。在保证实时性的前提下,系统操作站和控制站的数量可以从几个到几十个,可满足从几个回路、几十个输入输出(I/O)信息量到上千个控制回路、上万个 I/O 信息量的用户应用要求。

集散控制系统的组成　集散控制系统(DCS)的基本组成包括现场控制站、过程控制网络、操作员站、工程师站以及数据服务站等,如图 1 所示。

(1) 过程控制网络　适合于工业现场环境使用的高速通信网络,连接集散控制系统内各类功能站点的通信网络,它一般采用开放性和实时性较好的通信协议(如符合 IEEE802.3 的工业以太网协议)。为了保证控制系统的高可靠性、适应现场分散安装的特点,一般采用有一定安装间距、不同敷设方式的双冗余网络。同时为了保证系统完全性,该网络与其他网络信息(与企业信息管理网)交互往往安装安全通信网关或者防火墙。

(2) 工程师站　配有组态工具软件和系统管理软件的计算机。它为专业工程技术人员而设计,负责 DCS 的系统硬件配置、参数设定、控制算法编程

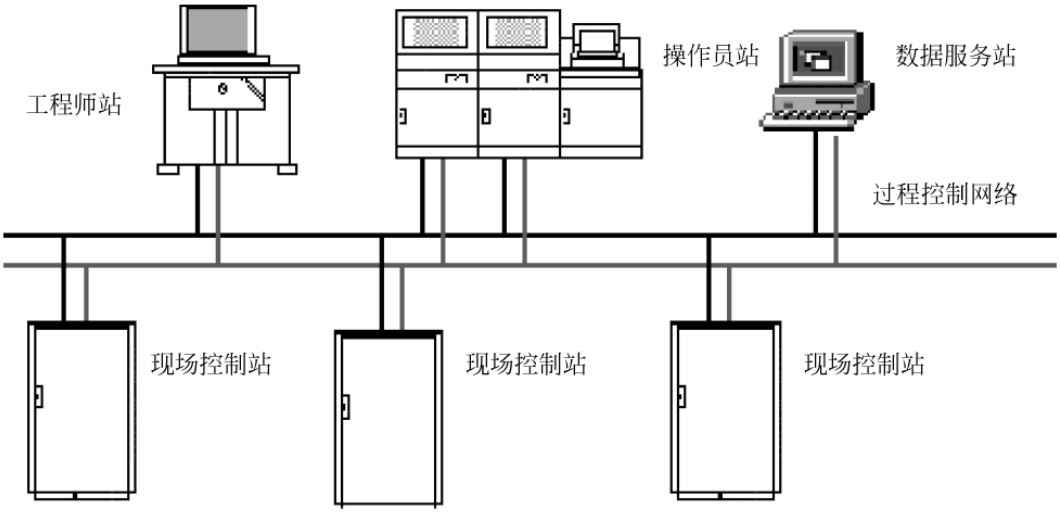


图 1　集散控制系统的基本组成

(如符合 IEC61131-3 的编程功能)、系统功能配置、数据库配置等组态工作,同时也负责系统诊断、维护和工程管理等功能。在许多系统中,工程师站还能提供用户进一步开发的高级语言软件,可完成有关工艺参数的复杂运算和数据分析工作,进行用组态软件无法实现的高级控制算法编程。

(3) 操作员站　操作工人完成过程监视和操作的人机接口设备,安装有专用的人机接口应用软件(HMI),主要用于以分级显示的画面(如总貌、分组、控制回路、动态工艺流程图等)监视生产过程运行的参数、状态和趋向,操纵生产过程,并可进行工艺数据的转储和打印等。

(4) 现场控制站　实现实时信号采集、运算及控制等功能,可通过工程师站软件组态的方式,灵活地满足不同的复杂控制要求,控制周期一般为 50 ~ 1 000 ms 不等,能实时进行循环执行并完成各种控制功能,现场控制站一般由高可靠性的嵌入式控制器和系列输入输出(I/O)模块组成,能实现模拟量输入、模拟量输出、数字量输入、数字量输出、脉冲量信号输入以及智能现场总线接口等。

(5) 数据服务站　负责过程数据的趋势、历史、报警、操作日志等信息的实时记录,往往采用高性能的服务器或者计算机,并配有实时数据库和管理软件,能实现不同规模的信号实时记录和管理。在系统规模较大的情况下,往往也可能单独配置组态数据服务站。

集散控制系统的软件　DCS 较有特色的功能软件包括:

(1) 组态工具软件　包括系统组态、过程控制组态、画面组态、报表组态等。所谓"组态",就是确定一些软件模块的连接关系,以生成相应的各种应用软件,使用户易于修改或制定新的控制方案。

(2) 实时数据库　应用于过程控制的实时信息交互的数据库。它必须满足实时控制要求的快速定期存储、实时处理、快速检索以及能长时间连续工作和信息安全的要求。处理的数据包括用户应用系统的组态信息、实时过程数据、过程数据报表、报警信息、历史数据等。组态生成的控制软件和人机接口软件都基于实时数据库运行。

(3) 实时控制软件　根据用户在工程师站的软件组态,经编译下载到现场控制站的主控制卡上的嵌入式软件,在系统内具有最高的实时性要求,运行周期一般为 0.01 ~ 1 s。

(4) 设备管理软件　利用现场总线(fieldbus)通信协议实现对现场变送器、阀门等智能设备管理,包括设备参数配置、设备诊断和故障报警管理、精度校准、操作日志和预测维护等。

集散控制系统的设计　为适应恶劣的工业现场环境(温度、湿度、电磁干扰、灰尘、腐蚀性气体、雷电、电网电压波动等)和控制要求,集散控制系统常采取以下措施:

(1) 高可靠性设计　DCS 的现场控制站和过程控制网络设备作为工业现场的电子设备,在电磁兼容性、机械结构、温度、湿度等环境指标上都应满足工业现场的要求。如采用所谓"三重隔离"(即通信接口隔离、电源隔离、I/O 通道隔离),更小的接地电阻,更高的隔离电压指标,固件密封等。此外,常要求有自诊断、自恢复或容错能力,并能容许对故障

模板带电插拔，结合冗余设计，显著减少维护时间。

（2）高实时性　为了保证控制性能，在 I/O 采集、过程数据通信等方面采用确定性通信方式保证系统数据采集和控制的高实时性。如采用高性能的微处理器、高速网络、例外报告等方法，用以提高系统实时性。

（3）高精度的信号快速采样　为了精确地反映并控制现场生产过程的工艺状态和产品质量，系统输入信号采集卡件和输出卡件一般要具备 0.1% ~ 0.5% 的精度，以及 100 ms ~ 1 s 之间的采集速度。

（4）输入输出（I/O）通道的本质安全性能　对于石油、化工等场合的应用，往往有易燃、易爆的气体或粉尘，因此对现场电气信号能量的泄放有严格的本质安全要求，输入输出通道按本质安全要求进行设计。

集散控制系统的发展方向　随着信息技术和现场总线技术的发展，集散控制系统向着开放、分散、智能、功能多样化的方向发展，DCS 目前为一种广义的概念，不仅指传统 DCS 厂家推出的新一代系统，也包括由可编程逻辑控制（PLC）、工业控制计算机（IPC）、高速总线网和专业厂家的组态软件所构成的系统。在不同的应用领域，DCS 的构成也各有不同。

（1）DCS 与现场总线（fieldbus）技术的融合　与现场设备（智能变送器、阀门定位器、变频器等）实现数字化通信，系统特性更加分散和智能。现场总线是一种双向、串行、多节点的全数字通信协议，目前常用现场总线协议包括 HART、FF、Profibus、ControlNet 等，实现现场智能设备与系统之间的数字通信。现场总线技术的应用可以增加传输信息的种类，提高传输速度和可靠性，减少布线费用。现场控制单元具有更好的稳定性和精度，能够实现更复杂的算法，更加分散化，实现更彻底的开放化。

（2）人机接口（HMI）软件的通用化和接口的统一　制定统一的软件层面的数据接口，可以提高硬件和软件的兼容能力，为客户提供更好的系统集成能力。OPC 软件协议的广泛使用使 HMI 软件的通用性大为提高，增加了工厂的综合自动化水平。

（3）工业以太网技术和系统功能综合化　工业以太网采用高速数据交换（switch）方式以及更高的电磁兼容性、环境适应性技术要求，统一解决了工业网络的纵向分层和横向远程的系统问题，使信息可以从设备传感器到管理层办公桌面完全集成。工业以太网高速传输性能和各个实时通信标准规约，使得控制系统功能和系统规模不断提高，通过控制分域、权限管理能互联和实现大规模联合生产装置的全局自动化，在保证信息和生产安全前提下实现全局信息共享。

（4）标准化和开放性　在系统功能和性能上指定了系列国际标准和国内标准，如控制编程标准（IEC61131-，包括 ST、LD、FBD、SFC 等编程规范）、现场总线标准（FF、HART、EPA、Profibus 等）、EMC 抗扰度测试标准（IEC61000，包括静电、浪涌、脉冲群等）、OPC 数据通信规约、安全标准（IEC61508）等，这些标准的应用使得各个制造厂家的系统具有更好的开放性。

参考文献

1. 凌志浩. DCS 与现场总线控制系统. 上海：华东理工大学出版社，2008

2. 刘翠玲，黄建兵. 集散控制系统. 北京：北京大学出版社，2006

3. 张雪申，叶西宁. 集散控制系统及其应用. 北京：机械工业出版社，2006　　（黄文君　王为民）

jihe zaoxing

几何造型（geometric modeling）　采用计算机进行几何形体建模的技术。几何造型技术是**计算机辅助设计**（CAD）、分析、制造以及**计算机图形学**的基础。几何造型技术自 20 世纪 60 年代出现以来，经过几十年的发展、进化，已日臻成熟，内容十分丰富。几何造型总体上可以分为线框造型、曲面造型和实体造型三类。

线框造型最简单，采用顶点和棱边表示几何形体（称为线框模型），通过交互输入几何形体的顶点和棱边进行几何造型。线框模型的特点是表示简单、处理算法高效。然而，由于采用线框模型表示三维物体时几何信息不完备，会产生二义性，并且无法支持对三维物体进行剖面图生成、隐藏线消除、面面求交等操作，因此，线框模型只适合用作三维形体的辅助表示。

曲面造型是应精确描述飞机、船舶、汽车的流线外形的需要而发展起来的，其核心内容是复杂曲面的数字化定义和表示方法。为曲面造型奠定理论基础的杰出先驱是法国的 P. Bézier 和美国的 S. Coons。Coons 提出了将分段曲线和分片曲面拼合成任意复杂曲线、曲面的一般处理方法。Bézier 则构想了另一种更加直观的曲线、曲面设计方法，即用折线来勾画曲线、曲面形状，生成 Bézier 曲线、曲面的方法。

1973 年 R. F. Riesenfeld 将 Bézier 曲面中的 Bernstein 基函数替换成 B 样条基函数，形成了 B 样条曲面。以后又进一步发展为非均匀有理 B 样条（NURBS）曲面。由于 NURBS 曲面能够统一表示平面、二次曲面、Bézier 曲面和 B 样条曲面，它已成为 CAD 系统中国际公认的产品外形定义方法。为了能够有效地构造任意拓扑曲面以及进行曲面的自适应局部细化，人们又提出了**细分曲面**和 T 样条曲面。上述曲面造型方法所生成的曲面基本都属于**参数曲面**。这些方法统称为自由曲面造型方法，其突出的优点是曲面构造直观，修改方便，能够构造任意复杂曲面等。隐式曲面造型是另一种曲面造型方法。**隐式曲面**易于进行位置关系确定、曲面求交运算，且在表示某些种类的几何形体时更加方便、准确，因此在计算机动画中的人体和动物造型方面获得了较好应用。具有代表性的隐式曲面造型方法是元球方法、卷积曲面方法等。除了自由曲面和隐式曲面之外，还有**网格曲面**和点云曲面。它们均是离散形式的曲面，随着三维扫描仪的广泛应用而受到重视，并在计算机图形学、逆向工程等方向得到很好应用。

实体造型旨在建立三维物体的完备几何模型，即实体模型，以充分满足计算辅助设计（CAD）、计算机辅助工程（CAE）、计算机辅助制造（CAM）应用的需要。实体造型技术产生于 20 世纪 60 年代末至 70 年代初，英国的 I. C. Braid 和美国的 A. A. G. Requicha 是该技术的主要奠基人。传统的实体模型表示方法主要包括边界表示（B-rep）、体素构造表示和空间枚举表示等。实体造型操作主要有拼合操作、扫成操作、欧拉操作、局部操作等。传统的实体造型经过 10 多年的演变进一步发展为基于参数化和特征的实体造型，其特点是用带有特定工程语义的形状特征取代传统的基本体素进行实体造型。由于形状特征为本行业的专业人员所熟悉，并带有特定的设计知识，因此基于特征的实体造型更符合设计人员的设计习惯，并且能够生成具有语义的实体模型。参数化实体造型方法的实质在于将几何约束引入实体模型中，形成以尺寸为参数的参数化实体模型，以有效支持变动设计。采用参数化实体造型方法，在构造出几何形体的参数化模型后，可以通过修改当前几何形体的尺寸参数值自动生成与新的尺寸相匹配的几何形体，从而显著提高变动设计效率。

随着工业界对几何造型技术的要求越来越高，需要进行几何建模的对象越来越复杂，几何造型技术自身还在不断发展。主要发展趋势包括：支持高度复杂几何模型的构建与处理、基于语义历史无关的直接几何建模、基于草绘和三维交互技术的便捷几何建模、基于图像的快速几何建模等。

参考文献

1. Mortenson M E. Geometric Modeling. 2nd ed. New York: John Wiley & Sons, 1997

2. Shah J J, Mäntylä M. Parametric and feature based CAD/CAM: concepts, techniques, and applications. New York: John Wiley & Sons, 1995

3. 唐荣锡，汪嘉业，彭群生，等. 计算机图形学教程. 修订版. 北京：科学出版社，2000

（高曙明）

jiliang yuyanxue

计量语言学（quantitative linguistics） 用概率论、**数理统计**、随机过程等数学方法研究各级语言单位的出现频率和分布规律的学科，目的是发现语言中存在的统计分布法则并进而形成具有量化特点的语言学理论。

当人们对语言和文本进行观察时，记下不同层次上各类语言单位（如音素，音位，字及其笔画、部件、结构方式，词，词类，词义，短语结构，句型等）的出现次数，并在所得的频率数据之间寻找相互关系，就可能发现某种支配这些单位在使用时的统计规律。语音、文字、词汇和语法的许多方面都曾经运用这种方式进行观察和研究，并获得了有实用价值的统计法则。这说明人们的语言行为严格地遵循着某些统计期望。调查表明某些统计规律甚至普遍适用于所有语言。美国哲学家 G. Zipf 提出的 Zipf 定律就是一条与语种无关的普遍法则。该定律认为在一种语言中每个词在按降序排列的词频表中的秩（又称词级）r 和它的出现频率 f 之间存在简单的反比关系。这一关系与文本的作者、主题或其他语言因素无关。我国学者在进行现代汉语词频统计时论述了这条定律同样适用于汉语，只是在频率最高和最低的那些词上略有差异。Zipf 通过观察还发现，一个词的长度与其出现频率成反比关系。换句话讲，越是频繁使用的词，其语音越简单且词长越短。现代汉语的词频统计同样验证了这条统计规律：在全部词中，汉语双音节词占 73.6%，单音节词只占 12%；但就词次而言，单音节词占 64%，双音节词占 34%，其余三音节以上词的词次总和不到 2%。G. Zipf 把这条统计规律解释为语言使用中的最省力原理。字频和词频统计在语文教学、字典和词典编纂以及作家的文章风格研究中所起的作用已经为人们

所熟知。它们对设计汉字编码(键盘)输入方案(参见**汉字键盘输入**)、信息交换用编码字符集、**汉字识别**、**汉语言语识别**、自动分词和文本校对等中文信息处理领域也有重要的贡献。

这个学科的主要研究内容包括：①运用精密的数学方法研究语言和文本的计量特性；②把计量语言学的方法、模型和研究成果应用于**自然语言处理**、**机器翻译**、信息检索、语言教学和文档生成等实用性课题；③把自然科学、经济学和心理学等学科中的科学方法和模型引入语言文字学。

国际学术界成立了国际计量语言学学会(IQLA),定期举行国际学术会议,并出版学报 *Journal of Quantitative Linguistics*。

参考文献

1. 陈原. 现代汉语定量分析. 上海：上海教育出版社,1989

2. Charniak E. Statistical language learning. Cambridge,MA：MIT Press，1993

3. Butler C. Statistics in linguistics. UK：Basil Blackwell,1985

(黄昌宁　常宝宝)

jisuan fuzaxing lilun

计算复杂性理论(computational complexity theory)　用数学方法研究各类问题的计算复杂性的学科。它研究各种可计算问题在计算过程中资源(如时间、空间等)的耗费情况,以及在不同计算模型下,使用不同类型资源和不同数量的资源时,各类问题复杂性的本质特性和相互关系。具体说,第一个层次是具体复杂性,它研究各种问题的时间和空间复杂度。这一类研究通常包括在算法设计与分析学科分支中。第二个层次是问题复杂度的上界和下界研究。虽然这种研究仍是针对具体问题的,但它包含了问题复杂性的一些本质特性,也采用了一些更具广泛意义的技术与方法。再上一层是研究语言复杂性类,研究它们的时间、空间以及时空折合等,研究确定性与非确定性对复杂度的影响。最后,研究完全性、相对性和相似性。

资源与复杂度函数　最重要的资源是计算过程所占的时间和所用的空间。资源的精确定义与所采用的计算模型有关。其他资源尚有图灵机的巡回,布尔线路的大小、深度和宽度,向量机器模型的并行时间等。计算过程所耗费的资源与问题的规模有关。例如,n 个数的排序,n 越大,所需的时间自然越多。比较好的排序算法的时间是 $cn \log n$(c 是一常数)。在计算复杂性理论中,算法所需的时间和空间都是问题规模 n 的函数 $T(n)$ 和 $S(n)$,它们分别称为时间复杂度函数和空间复杂度函数。当 n 确定以后,可能还有多种不同的实例,不同实例的计算时间仍可能不同。如果选用所有规模为 n 的实例所用的最长时间作为 $T(n)$,则称之为最坏情况时间复杂度。如果选用各实例所用时间的平均值作为$T(n)$,则称之为平均时间复杂度。大多数情况下,采用最坏情况复杂度。类似地可定义空间复杂度。

计算模型　最基本的模型是**图灵机**。作为时间复杂性分析的模型是*多带图灵机*。它有 k 条双向无限带,其基本动作是：根据有限控制器的状态和 k 条工作带带头读到的 k 个字符,执行三个操作,即 k 个带头各自写下一个字符,改变机器状态,k 个带头独立地左移或右移一格。对于每个长度为 n 的输入,图灵机 M 在停机前最多做 $T(n)$ 个动作,就称 M(及其对应的算法)的时间复杂度为 $T(n)$。作为空间复杂性分析的模型是*离线图灵机*。它有一条只读输入带和 k 条半无穷存储带。其优点是可以进一步考虑比线性空间更小的空间耗费,例如对数空间。对于每个长度为 n 的输入,图灵机 M 在任一条存储带上至多扫视 $S(n)$ 个方格,就称 M(及其对应的算法)的空间复杂度为 $S(n)$。

其他计算模型还有随机存取机(RAM),非确定图灵机、Oracle 机器、布尔线路以及并行计算模型 P-RAM和向量机器模型等。

上界与下界　研究计算复杂性的一个重要目的是要为各种问题寻求最优算法。这就要求知道问题的固有复杂度。目前对大多数问题而言,还只知道它最多需要多少、至少需要多少某种资源。以时间为例,前者称为该问题时间复杂度的上界,后者称为下界。上界越小越好,下界越大越好。一旦对某问题,其上、下界相等了,就知道了该问题的固有复杂度,具有这种复杂度的算法自然是最优算法。上、下界研究是针对单个具体问题的。

寻找问题的上界,没有固定方法。找到问题的一个算法,若其时间为 $T(n)$,则 $T(n)$ 即为该问题时间复杂度的一个上界,因为这表明,解决问题用 $T(n)$ 时间就足够了。更小的上界,取决于设计更好的算法。n 点离散傅里叶变换的时间上界为 $O(n \log n)$。两个 n 维矩阵的乘法,通常需要$O(n^3)$时间。用分治法设计的一个算法,使时间上界降到 $O(n^{2.81})$。经过一系列工作,时间上界已降到$O(n^{2.376})$。

寻找问题的下界，则困难得多。因为此时必须证明，该问题的一切算法（包括已知的和未知的），都至少需要这么多资源。证明下界的常用方法有计数论证法和归约方法。目前关于下界的结果，多集中在语言理论和逻辑理论中。下面是两个通常问题的下界结果：n 个数的排序，其基于数的比较的时间复杂度至少是 $n \log n$；为了在图灵机上判断一个长度为 n 的数是否是某数的完全平方，其时间和空间的乘积至少正比于 n^2。

语言复杂性类 语言（参见**形式语言理论**）是字符串的集合。如果对于一个输入字符串，图灵机操作到最后，进入某个接受状态并停机，就称这个图灵机接受该字符串。如果一个图灵机接受且只接受某个语言中的字符串，就称这个图灵机识别该语言。语言可以按识别它们时所耗费的时间，分成不同的语言复杂性类，例如，DTIME（n），DTIME（n^2），DTIME（n^3），…，DTIME（2^n），…。这里 DTIME（$f(n)$）表示确定型图灵机在 $f(n)$ 时间内所能识别的所有语言组成的类。类似地，语言也可按所耗费的空间分成各种语言复杂性类，例如DSPACE（$\log n$），DSPACE（n），DSPACE（n^2），…，DSPACE（2^n），…。还有两个重要的语言复杂性类 P 和 PSPACE，它们分别是在多项式时间和多项式空间内能识别的语言组成的类。显然，P 包含 DTIME（n），DTIME（n^2），…，DTIME（n^k），…。P 类问题（对应于 P 类中语言的问题，参见 **P 类问题**）都是多项式时间内可解决的问题，被公认为是在计算机上现实可解的问题，解决它们的多项式时间算法被认为是现实可行的算法。

计算复杂性理论的一些主要研究工作都集中在语言复杂性类上。这是因为各种问题的求解，各种函数的计算，都可转化成语言的识别。在计算复杂性理论中，问题由无数实例组成，实例经过编码可表成字符串，所有具有肯定回答的实例所对应的字符串就构成一个语言。因此，判定一个实例是否有肯定回答，就转化成判定一个字符串是否属于一个语言。对于函数的计算，可以在图灵机上增加一条输出带，自变量值经编码可作为输入字符串，设计图灵机时，保证当机器停机并进入接受状态时，输出带上正好写下了函数值。这样，也转化成语言识别问题。

时间谱系与空间谱系 某些基础研究工作在于建立语言复杂性类的时间和空间谱系，即要弄清楚语言复杂性类由于时间和空间等资源量的不同，是怎样一层一层扩大的。线性加速定理表明，时间和空间仅由 $f(n)$ 增加到 $kf(n)$，所识别的语言不会增加，因而语言复杂性类不会扩大。关于谱系的一些基本定理回答了时间和空间究竟要分别增长到什么程度，才会使所识别的语言有所增加。此外，还证明：任给一个资源函数 $f(n)$，总会有语言不在 DTIME（$f(n)$）（或 DSPACE（$f(n)$））中，因而时间谱系和空间谱系都是无穷的。

确定性与非确定性 一个确定型图灵机，对于其状态和所读到的字符，只有一个唯一的动作（包括改写字符、改变状态和移动带头等三个操作）。如果有多个动作可供选择，此时，由于每步都可能有几个不同的动作，这种图灵机对一个输入字符串，从开始到停机，将可能会有多种不同的动作序列，若规定只要有一个序列终止在接受状态，机器就接受这个输入字符串，则这种机器称为非确定型图灵机。已经证明，非确定型图灵机与确定型图灵机所识别的语言类是相同的。若一旦考虑有界的时间与空间，它们的识别能力需认真分析。

对于非确定型图灵机，也可按时间和空间界限，将语言分成许多语言复杂性类，形成非确定时间和空间谱系。例如，NTIME（n），NTIME（n^2），…，NTIME（2^n），…；NSPACE（$\log n$），NSPACE（n），…，NSPACE（2^n），…。这里也有两个重要的语言复杂性类 NP 和 NPSPACE，它们分别是非确定型图灵机在多项式时间和多项式空间内识别的语言组成的类。在确定型和非确定型语言类之间，显然有下面关系成立：DTIME（$f(n)$）$\subseteq$ NTIME（$f(n)$），DSPACE（$f(n)$）$\subseteq$NSPACE（$f(n)$）。在时间和空间之间有：DTIME（$f(n)$）$\subseteq$ DSPACE（$f(n)$），NTIME（$f(n)$）$\subseteq$NSPACE（$f(n)$）。有大量工作是集中在研究各种语言复杂性类之间的关系上的，研究时间与空间的关系、确定性与非确定性的关系。已经证明，DTIME（n）$\neq$NTIME（n）。这说明在线性时间界限内，确定型图灵机所识别的语言类要严格地小于非确定型图灵机所识别的语言类。另一方面，也已证明，PSPACE = NPSPACE。这说明在多项式空间界限内，确定型图灵机和非确定型图灵机所识别的语言类又是相同的。这个结论来自一个著名的定理：NSPACE（$S(n)$）$\subseteq$DSPACE（$S^2(n)$）。有许多问题还不明朗，例如是否有 DTIME（n^k）$\neq$NTIME（n^k），NTIME（n）= DSPACE（n）等。在确定性和非确定性关系方面，还有一个十分著名的悬案：NP =？P（相关的还有 NP =？PSPACE）。这是一个在计算机科学领域中历经多年尚未解决的问题。它在理论上和

实践中都有极其重要的意义,它的解决不仅将导致一系列理论问题的解决,还将关系到计算机科学、数学、逻辑学和运筹学中一大批实际问题是否是“现实可计算的”(参见 **NP 完全性理论**)。

完全性　在语言复杂性类中,常常要研究类中最困难的语言(最困难的问题),完全性就是最困难性的形式表示。利用归约,可以解决两个问题的困难程度的比较问题,从而有助于完全性的定义。有各种不同形式的归约。在研究 NP 语言复杂性类的完全性时,要利用多项式时间多一归约。一个语言 L 多项式时间多一归约到语言 L_0,如果存在一个确定型图灵机,它在多项式时间内将一个字符串 x 转变成一个字符串 y,并使得 x 是 L 的字符串,当且仅当 y 是 L_0 的字符串。于是 L 的识别问题就归结为 L_0 的识别问题,反之,L_0 体现了 L 的难度。一个语言 L 称为是 NP 完全的,如果 L 是 NP 中语言,且 NP 中任一语言都能多项式时间多一归约到 L。类似地可以定义 P 完全性,PSPACE 完全性等。由于 NP 类中一切语言的识别都可以归结为一个 NP 完全语言的识别,因此,可以认为 NP 完全语言是 NP 类中最难于识别的语言。在诸多的完全性研究中,NP 完全性研究有特殊地位,它包括许多研究课题,是计算复杂性理论中一个重要研究领域。其研究与 NP = ? P 问题有密切关系,在理论和实践两个方面,都有重要意义(参见 **NP 完全性理论**)。

相对性　这是从另一个角度来比较问题求解的复杂程度,它需要借助某种外部信息。研究的基本工具是 Oracle 机器。Oracle 机器包括一台图灵机,一个 Oracle 集(直译为神灵启示集或译为外部信息源),一条特殊带和三个特殊状态 Query,Yes 和 No。每当机器进入 Query 状态时,机器就要询问 Oracle 集:特殊带上写的字符串是否是 Oracle 集中的字符串。如果是,机器进入 Yes 状态,否则进入 No 状态。然后进入不同的后续操作。整个机器动作,除上述情况外,与图灵机无异。这是一种相对化的计算,它不仅依靠机器自身的计算,还强烈地依赖于 Oracle 集合的性质。Oracle 集合可以是一个具有某种复杂度的语言,甚至也可以是一个不可计算的集合。以 A 为 Oracle 集合的确定的 Oracle 机器在多项式时间接受的语言类,记作 P(A),类似地可定义 NP(A)。

利用 Oracle 机器,可以形成许多新的相对化的语言复杂性类,它们可以组成一个完整的体系——多项式谱系,这个谱系有着丰富的内容。在相对性研究中最重要的一个研究结果是:存在一个集合 A,使得 NP(A) = P(A),同时,还存在另一个集合 B,使得 NP(B) ≠ P(B)。这一结果更显示了解决 NP = ? P 问题的困难程度。

相似性　由于计算模型的不同,同一类问题所用的资源可能不尽相同。那么,这些复杂度函数之间究竟呈现什么关系呢?有人提出并行计算假设:并行机器的时间复杂度多项式相关于图灵机的空间复杂度。在全面深入研究的基础上,人们又提出相似性原理:对于同一类问题,各理想的计算模型使用本质上同样多的并行时间,本质上同样多的空间和本质上同样多的串行时间。“本质上同样多”的含义是多项式相关。对于图灵机,并行时间系指巡回。

此外,还有描述复杂性和 Kolmogorov 复杂性等研究方向。

目前,随着科学技术的发展,计算复杂性理论又进入一个新的发展阶段。日益占据重要地位的并行计算,推动了并行计算复杂性理论的发展。特别是人工智能和智能计算机的研究,由于涉及对人类智力行为的模拟,以及巨大的信息处理量,必将对计算复杂性理论提出崭新的研究课题。

参考文献

1. Hopcroft J E, Ullman J D. 自动机理论、语言和计算导引. 徐美瑞,译. 北京:科学出版社,1986

2. Arora S, Barak B. Computational complexity: a modern approach. Cambridge, UK: Cambridge University Press, 2009

3. Balcázar J L, Díaz J, Gabarro J. Structural Complexity Ⅰ, Ⅱ. Berlin: Springer-Verlag, 1988

4. Papadimitriou C H, Computational complexity. Boston, MA: Addison-Wesley, 1994

(徐美瑞)

jisuanji

计算机(computer)　主要用电子器件和电路,配以各种相关设备而组成的自动计算装置。通常所讲的计算机即电子计算机。根据信息或数据的表示方式和组成原理的不同,电子计算机大体上可分为电子数字计算机,简称**数字计算机**和电子模拟计算机,简称**模拟计算机**两大类。这两类计算机的根本差别在于:①前者是对采用离散的符号或数字表示的信息或数据进行计算;后者则是对用电流或电压等来表示的连续物理变量进行计算,以解算诸如微分方程组等问题。②前者是在存储程序的控制下,顺序执行计算所需的操作;后者则事先在控制面板

上排好解题程序,并设置好初始条件和有关参数,然后启动计算机,进行连续的运算。

历史上的计算工具

早在远古时代,人类就有了数的概念和对数进行简单计算的方法,随后渐渐发展为简单的计算工具。我国在春秋战国时期(公元前 8 世纪到公元前 3 世纪),“算筹”已被用来作为计算工具,并已得到广泛应用。“算盘”在南北朝的著作(约公元 570 年)中已有记载,到元代(公元 1271—1368 年)已普及应用, 而且沿用至今。应当提到的是:“二进制位”的概念起源于中国。《易经系辞》(约公元前 3 世纪初)中表示“阴”和“阳”的符号是“爻”。“爻”有两个符号,即“阳爻”“——”和“阴爻”“— —”。因此,一个“爻”便可以被认为是一个二进制的“位”。易经中的八卦是用 3 个爻表示,而六十四卦是用 6 个爻表示。德国数学家和哲学家莱布尼茨(Gottfried Wilhelm Leibniz)于 1679 年发表了关于二进制数的论文,但他在研究易经后,于 1716 年的著作中说:“伏羲氏在其推演的八卦中使用了二进制算术。”从 17 世纪开始,机械式的计算工具在欧洲得到了发展。例如:法国数学家和物理学家, 布莱斯帕斯卡(Blaise Pascal)在 1642 年创造了一种用齿轮驱动的十进制加法器;莱布尼茨在 1674 年研制了可进行十进制四则运算的手摇计算机等。英国的数学家和哲学家巴贝奇(Charles Babbage)是可编程计算机概念的发明者,他从 1822 年开始设计和制作“差分机”,这是一种可按事先规定的步骤计算多项式的机械式计算机,他在 1832 年制作了一个部件后,就转向设计可在穿孔卡片的控制下自动进行计算的“分析机”。由于种种原因,分析机未能制成。

从 19 世纪中叶到 20 世纪前期,有不少发明和创造性的工作,其中一些对后来电子计算机的发明和发展起着先导作用。例如布尔(George Boole)于 1854 年把逻辑归结为一种命题演算,提出了符号逻辑运算系统,并认为它可以作为计算机的设计基础,这就是后来获得广泛应用的**布尔代数**。在这个时期,计算机的发展从机械式转向机电式。1884 年第一个机电式加法机出现。1889 年建造出电气制表系统。1936 年,英国数学家图灵(Alan Turing)提出图灵机的数学模型。1938—1944 年,德国的宙瑟(Konrad Zuse)先后研制了 Z2、Z3、Z4 等二进制机电式过程控制数字计算机。艾肯(Howard Hathaway Aiken)在 1944 年到 1947 年间在美国哈佛大学成功研制出用继电器组成的自动顺序控制计算机 MARK-Ⅰ和 MARK-Ⅱ。

模拟计算工具的出现可以追溯到公元前数百年。代表性的模拟计算工具有 17 世纪 30 年代发明的圆周形计算尺和 1850 年的条形对数计算尺。对数计算尺适合于乘、除法和多种常用函数的计算,而且结构轻巧,便于携带, 因此在随后的 100 余年里, 成为工程设计和科学实验人员的常备计算工具。19 世纪中期至 20 世纪 40 年代, 也是模拟计算机从机械式到机电式的发展时期。例如,1930 年在美国 MIT 设计制作了机电式的微分分析机,用来求解各种微分方程。

电子计算机的诞生

1942 年,美国 John Vincent Atanasoff 等研制了一台二进制的电子管数字计算机原型, 用于复杂而费时的理论物理计算。

1943 年,美国宾夕法尼亚大学莫尔学院的 John Presper Eckert Jr 和 John Willianm Mauchly 研制用电子管组成的电子数字计算机 ENIAC, 该机于 1946 年 2 月投入运行,并公之于世。当时它是一台前所未有的、复杂而庞大的电子机器,用了18 000多支电子管,重量达 30 t, 机房面积约 140 m^2, 并达到了其他数字计算机望尘莫及的运算速度, 即每秒运算 5 000次 10 位十进制数的加、减法运算。ENIAC 开创了电子数字计算机蓬勃发展的新纪元。

1945—1946 年,冯·诺依曼等提出了“存储程序”的概念和采用二进制逻辑与存储过程控制的电子数字计算机的基本设计思想和体系结构, 即所谓的冯·诺依曼结构(参见**数字计算机**)。

从 20 世纪 50 年代初期开始,陆续推出了商品化的计算机。在 20 世纪后半个世纪到 21 世纪,电子数字计算机得到了极大的发展,从最初的电子管计算机到其后的晶体管计算机、集成电路计算机和超大规模集成电路计算机。

在模拟计算机方面,自从 1952 年电子管的运算放大器投入市场后, 电子模拟计算机及其应用开始迅速发展。早在电子数字计算机问世之前的第二次世界大战期间,模拟计算机就已经在武器的实时跟踪瞄准和射击控制等方面得到应用。此后又在某些领域,例如工程设计的动态模拟、生产过程控制、动态系统仿真等,获得了广泛应用。在 20 世纪 50 年代后期,出现了把数字计算机和模拟计算机结合起来组成混合计算机。此外,还有一种用数字积分器来取代模拟计算机中运算放大器的数字微分分析机。由于数字计算机的高度发展和广泛的普及应

用,在许多传统上以模拟计算机为主的应用系统中,往往可以更好地用数字计算机来取代,所以通常所说的电子计算机或计算机指的就是电子数字计算机。

计算机的应用

英国科学家图灵对于计算的研究和图灵机模型的概念提出以及美籍匈牙利科学家冯·诺依曼关于存储程序计算机和冯·诺依曼计算机结构不但奠定了计算机系统发展的基础,而且为计算机的广泛应用奠定了基础。最初的数字计算机的应用领域是科学和工程计算,其后的发展包括了控制、商务、军事、教育、通信以及社会事务的各个方面。现在,计算机已经在人类社会的各个领域获得了广泛的应用。计算机应用的类型主要有 3 种:计算密集型、数据密集型和通信密集型。计算密集型应用包括大规模科学和工程计算以及数值模拟等,数据密集型应用包括数字化图书馆、数据仓库、计算结果可视化等,通信密集型应用包括计算机支持的协同工作、遥控、远程医疗等。

计算机应用领域举例如下:

(1) 生物学、生物化学和医学　药物设计、生物分子结构、基因信息学、医学图像处理、人体仿真、医学数据库分析、医疗管理系统等。

(2) 化学和化工　催化剂和酶的研究与设计、化工厂管道系统的设计等。

(3) 物理学　材料性质研究、芯片中的半导体模拟、结构力学计算和性能模拟、流体动力学计算、核聚变和核爆炸模拟、基本粒子性质研究等。

(4) 空间科学和天文学　宇宙形成研究、银河系形成研究、太阳动力学、黑洞和引力波研究、太空探测数据处理等。

(5) 人工智能　人工神经网络学习和优化算法、数据库检索、计算机下棋等。

(6) 天气和气象　天气预报和气象预测、灾害性天气预报等。

(7) 环境科学　污染物和地下水在地层中的流动模型、地球生态环境模拟、根据卫星遥测数据研究土地资源的性质和利用等。

(8) 地球物理和石油工程　石油勘探中的三维地震资料处理、油藏模拟、地震预报研究等。

(9) 航空、航天、机械和制造工业　汽车和飞行器的设计和制造、推进器设计、飞行器的雷达成像、计算机中的芯片模拟、高频电路的电磁特性模拟、生产过程模拟、制造系统的优化调度等。

(10) 军事应用　武器设计、军用传感器和通信系统模拟、军事决策支持系统、军事演习模拟等。

(11) 商业和事业系统　非正常情况(天气异常或人为事故)下空中交通的动态调度、机器人的控制和图像分析、数字化电影中的图形学、经济模型、电子商务、税务系统、商业决策支持系统、银行业务支持系统、信用卡和股市信息支持系统等。

(12) 消费电子领域　数字电视、数码照相机、计算器、数字事务助理(PDA)、电子词典、电子游戏、电子书、数字音视频播放器(MP3、MP4)、全球定位系统(GPS)等。

(13) 通信和信息网络　计算机通信、因特网、万维网、视频会议、远程教育、数字化图书馆、社会保障信息支持系统等。

展　望

电子计算机对经济发展、社会进步和科学研究将起到越来越重要的作用,它将使人类进入更高层次的信息化社会。高性能计算机的发展将使过去一些难以解决的极其复杂的重大问题获得解决。**网格计算**将分布在不同地理位置和不同类型的计算资源通过互连网络连接起来,实现资源共享和协同工作。用户使用网络资源时,不需要知道资源的提供者及其所在的地理位置。移动式计算(参见**移动式计算机**)可使用户通过互连网络在任何地点和任何时间进行计算。嵌入式计算机(参见**嵌入式计算机**)可直接装入机电设备、仪器仪表或家用电器内部,它们将在人们的周围环境中存在而不为使用者所觉察,它们无处不在,无时不在。云计算的发展使得计算可以不是以某种产品的形式(软件的、硬件的)而是以服务的形式提供给用户,可以使用户更方便地共享计算资源,为计算机的应用提供更加广阔的空间。

对于计算机所用的器件,自从使用半导体微电子技术以后,集成电路芯片的集成度(即单位面积所能集成的晶体管数量)基本上是按照摩尔定律增长的,即每隔 18 个月左右,集成电路的集成度增加 1 倍,其性能也增加 1 倍。然而在 21 世纪初期,硅半导体大规模集成电路技术已发展到接近于它的物理极限。为了进一步提高计算机的性能,除了在计算机体系结构方面将计算机组成高度并行和分布式的系统外,还将开发新型器件。例如纳米器件、量子器件、具有量子效应的超导器件、光调制器件、光-电子集成器件、分子器件、生物分子器件等。此外,利用光计算、量子计算和生物计算的原理可以制造光计算机(参见“**光计算机**”)、量子计算机(参见“**量**

子计算机”)和生物计算机(参见“**生物计算机**”)等新型计算机,利用它们有可能组成新型的高性能计算机或高性能协处理机。

总之,现代电子计算机历经半个多世纪的发展,已经成为当今文明社会须臾不可或缺的、最得力、最灵活、最普遍的计算以及通信和信息处理工具,今后仍将继续沿着提高其性能、强化其功能、增加其易用性和灵活性、进一步拓广和深化其应用等方面不断发展。电子计算机的前景是不可限量的。

参考文献

1. Ralston A, Edwin E D, ed. Encyclopedia of computer science. 3rd ed. N. Y.: IEEE Press, 1993

2. Saha D, Mukherjee A. Pervasive Computing: A Paradigm for the 21st Century. Computer,2003, 36(3): 25-31

(童频　刘志勇)

jisuanji bingdu

计算机病毒(computer virus)　编制或者在计算机程序中插入的破坏计算机功能或者毁坏数据,影响计算机使用,能自我复制的一组计算机指令或者程序代码。它是一种特殊的计算机程序,能够自我复制,寻找宿主对象,并且依附于宿主,但不主动传播。

计算机病毒具有传染性、依附寄生性、隐蔽性、潜伏性、欺骗性、可执行性、可触发性、可复制性、顽固性、破坏性等特点,对计算机系统与网络的安全和正常运行具有极大的危害性。

计算机病毒的病理机制与人体感染细菌和病毒的病理现象十分相似,能通过修改或自我复制向其他程序扩散(传染),进而扰乱系统及用户程序的正常运行。

病毒程序发作前寄生在介质上的某个程序中,处于静止状态。一旦带病毒程序被引导或调用,它就被激活,变成有感染力的动态病毒。当传染条件满足时,病毒就侵入内存,随着作业进程的发展,逐步(有时很快地)向其他作业模块扩散,并传染给其他软件。在破坏条件满足时,它就由表现模块或破坏模块把病毒以特定的方式表现出来,造成计算机软硬件破坏、数据破坏或干扰系统的正常运行。

病毒的主要表现现象有:计算机经常性无缘无故地死机、操作系统无法正常启动、运行速度明显变慢、软件经常发生内存不足的错误、打印和通信发生异常、经常要求对软盘进行写操作、应用程序经常发生死机或者非法错误、系统文件的属性发生变化、word 等文件另存时只能以模板方式保存、磁盘空间迅速减少、网络驱动器卷或共享目录无法调用、基本内存发生变化等。

病毒可分为文件型病毒、存储器驻留病毒、引导型病毒、隐蔽病毒、多态病毒、宏病毒和网络病毒等不同的类型。

病毒检查方法主要有比较法、搜索法、特征识别法、分析法,病毒防护技术主要有通用解密、人工智能、监控病毒源、主动内核、分布式处理和数字免疫技术等。

反病毒软件已发展了以下 4 代:简单的扫描程序、启发式的扫描程序、行为陷阱和全方位的保护。还有两种更为先进的反病毒方法是类属解密技术和数字免疫系统。类属解密技术使得反病毒程序可以容易地检测出复杂的多形病毒,同时保持快速的扫描速度。当包含了一个多形病毒的文件在执行时,病毒必须解密自身来激活。数字免疫系统是一个防止病毒侵犯的综合方法,它的成功依赖于病毒分析机制检测新的病毒血统的能力,通过持续地分析和监视找到最新病毒来连续更新数字免疫系统。

病毒的防护要从防毒、查毒、解毒三个方面入手,重视培养安全防范意识、切断病毒传播途径、消除病毒载体、安装防病毒软件定期对计算机进行病毒检测、建立多级病毒防护体系等。

参考文献

张仁斌,等. 计算机病毒与反病毒技术. 北京:清华大学出版社,2012

(刘宝旭)

jisuanji daishu

计算机代数(computer algebra)　研究代数算法的设计、分析、实现及其应用的学科。代数算法有简洁的形式化描述,有基于数学理论的严格证明。代数算法在计算机上用软件实现时,输入和输出都是代数符号表示。计算机代数处理非数值计算。计算机代数又称为符号计算或公式推演。

计算机代数包括各类数学问题的解的算法——从初等数学(如初等代数式化简、初等数论)到高等数学(如解微分方程)各类问题的求解算法,以及这些算法的具体实现和在各学科中的应用。

化简程序　化简就是利用等价关系进行变换,将集合 S 上的元素变换为 S 上的另一个“简单”的元素。设 T 是变换,~ 是集合 S 上的等价关系,$u \in S$,则这个化简可用

$$T(u) \sim u \quad \text{和} \quad T(u) \leqslant u$$

来描述,其中≤是描述"简单"性的偏序。在不同场合,它有不同的含义。例如,它可以是"$T(u)$比 u 的字符少","求 $T(u)$的值比求 u 的值容易","$T(u)$比 u 更容易理解","$T(u)$比 u 更便于下一步使用"等等。变换 T 称为准化简器。两个恒等的表达式经准化简器化简后所得形式可能不一样,即可以是 S 内不同的元素。为了使得两个恒等的表达式经化简后变为同一形式,即 S 的同一元素,应使用规范化简器 T,它可描述如下:对任一 u,$u \in S$

$$T(u) \sim u,\ T(u) \leqslant u$$

$$u \sim v \Leftrightarrow T(u) = T(v)$$

这样,S 内所有与 u 等价的元素有唯一的元素 $T(u)$与之对应。$T(u)$代表了一个等价类,称为 u 的规范形式。

代数扩域法　近世代数的理论已经证明:F 的代数扩域 $F(\alpha)$与多项式环 $F[\alpha]$是同构的,而且代数扩域 $F(\alpha)$的元素都可用多项式环 $F[\alpha]$的元素

$$a_n\alpha^n + a_{n-1}\alpha^{n-1} + \cdots + a_0$$

来表示,且这种表示的约简形式是唯一的。同样代数扩域 $F(\alpha_1, \alpha_2, \cdots, \alpha_m)$的元素也可以用多项式环 $F[\alpha_1, \cdots, \alpha_m]$的元素表示。欲化简一个表达式 S,先构造一个代数扩域 E,使得 $S \in E$;然后在 E 上化简 S,这就是代数扩域法。

多项式无平方分解　如果 $p(x)$没有重根,则称为无平方多项式。设 $p(x) \in R[x]$,$p(x)$的无平方分解是

$$p(x) = p_1(x)p_2^2(x)\cdots p_k^k(x)$$

其中,$p_k(x) \neq 1$,$p_i(x)$都是无平方多项式,且当 $i \neq j$ 时,$p_i(x)$和 $p_j(x)$互素。推演无平方分解式的算法称为无平方分解算法。

Berlekamp 算法　在 Z_p(其中 p 是素数)上将一元多项式因式分解的算法,其时间复杂度是 $O(n^3 + prn^2)$,其中 r 是因子数,n 是多项式的次数。

中国剩余定理　设 $p_0, p_1 \cdots, p_{n-1}$是两两互素自然数,$p = p_0 \times p_1 \times \cdots \times p_{n-1}$,则任一小于 p 的非负整数 u,$0 \leqslant u < p$,可用唯一一组剩余 $u_0, u_1, \cdots, u_{n-1}$表示,其中

$$u_i = u \bmod p_i, \quad i = 0, 1, \cdots, n-1$$

给定 $p_0, p_1, \cdots, p_{n-1}$时,可记作

$$u \leftrightarrow (u_0, u_1, \cdots, u_{n-1})$$

反之,若已知$(u_0, u_1, \cdots, u_{n-1})$且 $0 \leqslant u_i < p_i$,　$i = 0, 1, \cdots, n-1$,则

$$u = \sum_{i=0}^{n-1} c_i d_i u_i \bmod p_i$$

其中 $c_i = p / p_i$,$d_i c_i = 1 \bmod p_i$。研究求 u 的快速算法是计算机代数的重要课题。

符号微分　函数的导函数和偏导函数称为符号导函数,但习惯上常称为符号微分。根据数学公式可设计计算符号微分的递归算法。

符号积分　函数的不定积分、参变积分称为符号积分。符号积分是一个比符号微分困难得多的问题。困难在于:有的函数,如 e^{x^2},原函数存在,但又从理论上证明了它没有闭形式解,通俗地说就是"积不出来"。

计算机代数语言　比较有影响的有 ALTRAN,CAMAL,FORMAL,MACSYMA,muMATH,REDUCE,SAC-2,SCRATCHPAD,MATHEMATICA。我国自己研制的有 CASC 等。

计算机代数有着广泛的应用,应用范围从纯数学到工业部门。例如,它已用于数论、射影几何、特殊函数、微分方程、天体力学、相对论、高能物理、理论物理、等离子体物理、电动力学、流体力学、理论化学、教育等学科。也已用于船体设计、直升飞机设计、电子显微镜的设计、机器人的手臂运动、轴采矿工厂的设计等等工业部门。

参考文献

1. Buchberger B, et al. Computer algebra: symbolic and algebraic computation. New York: Springer-Verlag:1983

2. 袁仁保. 计算机代数. 长沙: 国防科技大学出版社,1989

3. 袁仁保. 符号计算语言 CASC 及编译实现. 南昌: 江西高校出版社,1993

(袁仁保)

jisuanji dianyuan

计算机电源(power supply for computer)

为计算机主机、外围设备供配电的设备。计算机对电源除了容量和质量的要求外,还有系统控制、检测和保护等要求。由于应用环境的不同,对计算机电源的需求各不相同,下面重点阐述大中型计算机电源和便携式计算机电源。

大中型计算机电源

大中型计算机电源除了电源设备外,还专门设有供配电和监控装置,把众多的交、直流电源联成一体,以保证计算机安全可靠地运行。通常在交流电网的输入端安装**不间断电源**(UPS),在正常工作状态下,它对输出的交流起稳定电压、稳定频率和抗干扰的作用;在交流电网发生故障时,它能向负载继续

供电一段时间。大中型计算机直流供电有两种方式:一种是将交流电网的交流电变换成50 V直流总线电压,然后用直流-直流(DC-DC)模块将总线电压变换成所需要的直流电压。直流总线和蓄电池相并联,当电网发生故障时,由蓄电池供电。另一种方式是把交流电压送到计算机的各个部分,再就地转换成所需要的直流电压。

直流电源有磁放大器电源、整流电源、线性电源、开关电源等,最常见的是线性电源和开关电源。

便携式计算机电源

便携式计算机用电源,又称移动电源是一个集储电,升压,充电管理于一体的便携式设备。“移动电源”这个概念是随着便携式计算机和数码产品的普及、快速增长而发展起来的,实际上是一种方便、易携带的随身电源。便携式计算机电源可分为内置式(它嵌入在计算机或数码产品的内部)和外接式(它独立于内置式产品,可为内置式产品充电,俗称“充电宝”)两类。不论是内置式还是外接式移动电源其工作原理和组成部分相同,由储电、升压和充电管理三部分构成。

储电 储电介质一般采用可充电锂电电芯(电池),因为该类电芯体积相对小巧,容量大,市场流通广,价格适中,被广泛用于移动电源产品。锂电的电压在2.7～4.2 V之间,电压随着电量的下降而下降。带内置式锂电的设备,充满电后,用内置电池供电,当锂电池电压降到设定电压,如2.7 V,必须再次充电。从充满电到必须再次充电之间的有效工作时间称该产品的续航时间。续航时间与该锂电池的容量,设备的工作状态等因素有关。充电电流过大,电能耗尽后未及时充电均会影响锂电池的寿命。

常用锂电电芯有聚合物锂电池和18 650锂电池。聚合物锂电池:标准电压3.7 V,形状较多,可定制,具有容量大,物美价廉,应用广泛等特点。18 650锂电池,指电池直径为18 mm,长度为65 mm,圆柱体型的电池:标准电压3.7 V,体积小,容量大,主要用在笔记本计算机等高端产品。

升压 2.7～4.2 V的锂电池电压是不能直接给内置锂电的产品充电的,因为内置移动电源产品的锂电电压也是2.7～4.2 V,同电位的电压之间是不能充电的。所以移动电源向外输出电能必须具有升压系统,把2.7～4.2 V的锂电电压升压到5 V,这样就可以给其他内置锂电的产品充电了,如**笔记本计算机**、**平板计算机**、数字相机、手机等。

升压主流技术基本采用直流到直流(DC to DC)的升压方式。也有降压方案,虽然效率比较高,但对电芯的一致性要求高,所以几乎很少采用。

充电管理 移动电源不是一次性设备,它可以反复使用数百次以上。当移动电源电能使用完后,必须给移动电源充电。所以移动电源内部还必须有充电管理系统。充电管理系统能根据锂电的电压,自动调节充电电流。过程有:预充,恒压充电,和浮充充电等。目前充电管理系统比较成熟,智能IC监控整个充电过程。主流管理IC有PT 4056, PX 40等。给内置锂电池的产品充电,可以用该产品配套的由交流市电供电的适配器来充电,在不便连接交流市电时,也可以用该产品配套的外接式移动电源来充电。外接式移动电源内储的锂电能量耗尽后,也可以用配套的适配器来充电。当对锂电池进行充电时,电池的正极上有锂离子生成,生成的锂离子经过电解液运动到负极。而作为负极的碳呈层状结构,它有很多微孔,达到负极的锂离子就嵌入到碳层的微孔中,嵌入的锂离子越多,充电容量越高。同样,当对电池进行放电时(即我们使用电池的过程),嵌在负极碳层中的锂离子脱出,又运动回正极。回正极的锂离子越多,放电容量越高。

移动电源的主要特性有:转换效率、移动性、通用性、安全性等。

移动电源的转换效率是衡量移动电源质量的主要因素,优质的移动电源能量转换率可达85%,普通的则为70%至75%,差的甚至只有30%,如转换率低于75%则意味着移动电源线路损耗较大,电芯本身的发热量也大,容易造成安全隐患。

移动电源的移动性是指移动电源体积小,容量大,重量轻、方便携带。

移动电源的通用性是指产品能够适用于最大范围的便携式计算机或数码产品。即移动电源能够为笔记本计算机、平板计算机、手机、PDA、数字相机、CD播放机等多种数码产品服务。

移动电源的安全性一直是人们关注的问题,移动电源一定要具备:短路、过充、过放、恒流、恒压等保护措施,还应有高性能电源管理技术。

随着移动互联网和移动计算机及移动终端的普及,以锂电池为核心介质的移动电源的年产销量已超过数亿台。移动电源通过技术创新,产品沿着标准化、系列化,容量更大、更轻薄、能量转换率更高、更廉价和更安全可靠的方向发展。

(方资端 韩承德)

jisuanji donghua

计算机动画(computer animation)　用计算机生成一系列可供实时播放的连续画面的技术。它可辅助传统卡通动画片的制作,也可通过对三维空间中虚拟摄像机、光源及物体运动和变化(形状、色彩等)的描述,逼真地模拟客观世界中真实的或虚构的三维场景随时间而演变的过程。所生成的一系列画面可在显示屏上动态演示,也可将它们记录在电影胶片上或转换成视频信息输出到录像带上。

计算机动画的研究始于20世纪60年代初,1963年美国AT&T Bell实验室制作了第一部计算机动画片。在80年代之前,主要集中于二维动画系统的研制,应用于教学演示和辅助传统的动画片制作。三维计算机动画的研究始于70年代初,至80年代中后期,由于具有实时处理能力的超级图形工作站的出现,三维**几何造型**技术和**真实感图形生成**技术取得很大进展,才促进了具有高度逼真效果的三维计算机动画技术迅速发展,并达到实用化、商品化的地步。到90年代,将计算机动画技术应用于电影特技取得了显著成就。在《魔鬼终结者》《侏罗纪公园》《玩具总动员》《泰坦尼克号》《变形金刚》《2012》等电影中,人们可以充分领略计算机动画的高超魅力。

动画的原理是利用人的视觉暂留特性使连续播放的静态画面相互衔接而形成动态效果。计算机制作动画的过程是用绘制程序生成一系列的景物画面,其中当前帧画面是对前一帧画面的部分修改。动画是运动中的艺术,正如动画大师John Halas所说,运动是动画的要素。一般来说,计算机动画中的运动包括:①景物位置、方向、大小和形状的变化;②虚拟摄像机的运动;③景物表面纹理、色彩的变化。计算机动画所生成的是一个虚拟的世界,虚拟景物可以是商标、汽车、建筑物、人体、分子、桥梁、云彩、山脉、恐龙或昆虫等,虚拟景物并不需要真正去建造,物体、虚拟摄像机的运动也不会受到什么限制,动画师可以随心所欲地创造他的虚幻世界。

计算机动画的制作主要包含如下步骤:①创意　根据设计的需要,由导演设计好动画制作的脚本;②预处理　扫描外部图像,输入外部资料;③场景造型;④设定材质和光源;⑤设置动画;⑥运动图像的绘制;⑦动画播放;⑧后处理;⑨动画的录制;⑩配音(包括背景音乐和台词)。

后处理在计算机动画中占据非常重要的位置。后处理是指动画后期的非线性编辑合成技术,包括蓝屏抠像、合成、图像形态变换、特殊光效等。理论上,它不属于计算机动画的范畴,但它是动画制作中必不可少的过程。特别是视频技术的数字化趋势,使该技术日益受到人们的重视。目前,已有许多优秀的后期合成软件。在这些软件中,大量采用了图像处理和**计算机视觉**技术(如摄像机反求),可方便地进行特技效果的处理和运动跟踪、深度合成等复杂操作。

计算机动画主要研究运动控制技术以及与动画有关的造型、绘制、合成等技术。例如:①动画形体造型技术;②动画运动控制和描述;③动画图像绘制技术和算法;④动态模拟、动画系统的集成环境;⑤关节体、人体动画;⑥动画语言与系统;⑦用于动画运动控制和生成的专门硬件设备及接口;⑧特殊视觉效果生成技术。

动画技术有很多,要对它们进行细致的分类是困难的,大致可分为**关键帧动画**、渐变和**变形动画**、**过程动画**、关节动画和**人体动画**、**基于物理的动画**等。计算机动画技术的成熟推动了动画软件的发展,目前有大量优秀的商用动画软件可供选用。

计算机动画可广泛应用于电影特技、商业广告、电视片头、动画片、游戏,网页设计、计算机辅助教育、军事演习、建筑设计、飞行模拟等。

参考文献

1. Rick Parent. Computer Animation: Algorithms and Techniques. 2nd ed. San Fransisco, CA: Morgan Kaufmann, 2007

2. Isaac V Kerlow. The Art of 3D Computer Animation and Effects. 4th ed. Wiley Publishing, 2009

(金小刚　王裕国)

jisuanji fangzhen

计算机仿真(computer simulation)　利用计算机建立、校验、运行实际(或假想)系统的模型以得到模型的行为特性,从而达到分析、研究该系统之目的的过程、方法和技术。这里的“系统”是广义的,它包括工程系统,如电气系统、热力系统、交通系统、机电系统、计算机系统等,也包括非工程系统,如社会系统、生态系统、经济系统等。

计算机仿真的基本活动　现代仿真技术大多是在计算机支持下进行的。计算机仿真有三个基本活动:即建立实际的或设想的系统模型(系统建模)、实现模型在计算机上执行(模型执行)、分析模型执行的输出结果(模型分析)。联系这三个活动的是

系统仿真的三要素,即系统、模型、计算机(包括硬件和软件)。它们的关系可用图 1 描述。

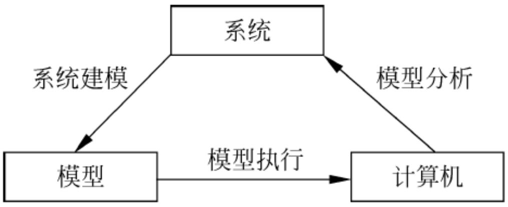


图 1　计算机仿真三要素及三个基本活动

现代仿真科学与技术是将仿真活动扩展到上述三个方面,并将其统一到同一环境中。

(1) 系统建模活动　除了传统的基于物理学、化学、生物学、社会学等基本定律及系统辨识等方法外,现代仿真技术广泛采用计算机辅助建模,产生了模型驱动框架(MDA,model-driven architecture),采用面向对象建模(object-oriented modeling)方法,在类库的基础上实现模型拼合与重用的基于模型库的结构化建模、基于元模型(meta-model)的建模等技术;还提出了用仿真方法确定实际系统的模型,例如,根据某一系统在试验中所获得的输入/输出数据,在计算机上进行仿真试验,确定模型的结构和参数。此外,为适应计算机软硬件环境的发展,许多新的建模方法在不断研究和开发出来,并取得了显著的进步。

(2) 模型执行活动　包括仿真建模与仿真实验。针对不同形式的系统模型研究其求解算法,使其在计算机上得以实现,进行"仿真程序"的检验(verification),这也称为"仿真建模"。现代仿真技术采用模型与实验分离技术,即模型的数据驱动(data driven)。它将实验框架与系统建模区分开来,模型分为两部分:参数模型和参数值。参数模型在系统建模活动中定义,实验框架定义一组条件,包括模型参数、输入变量、观测变量、初始条件、终止条件;参数值属于实验框架的内容之一。这样,模型参数与其对应的参数模型分离开来。仿真执行时,只需对参数模型赋予具体参数值,就形成了一个特定的模型,从而大大提高了仿真的灵活性和运行效率。

(3) 模型分析活动　现代仿真技术将输出函数的定义也从模型、实验框架中分离出来。这样,当需要不同形式的输出时,不必重新修改仿真模型和实验框架,甚至不必重新仿真运行,这样就能更方便地进行模型分析。模型分析还包括将仿真执行结果与实际系统的行为进行比较以便确认模型(validation)。

计算机仿真的一般步骤　计算机仿真的一般步骤可用图 2 来描述。

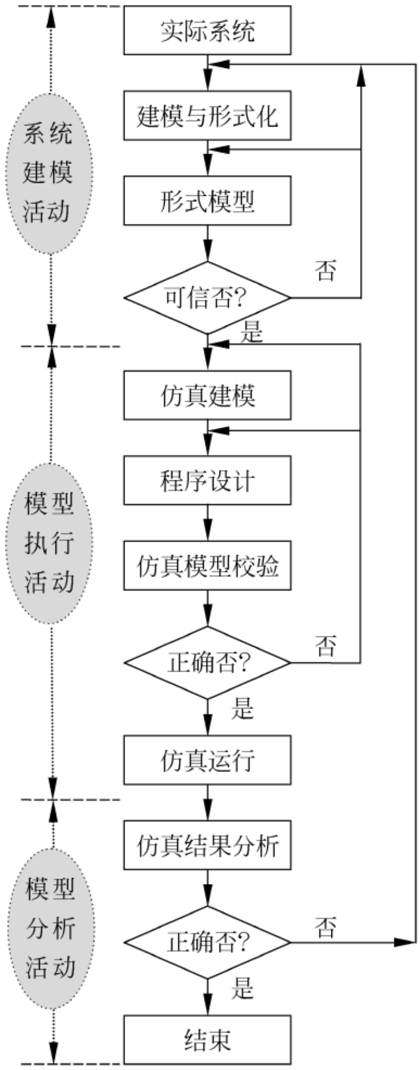


图 2　计算机仿真的一般步骤

首先要针对实际系统建立其模型。建模与形式化的任务是:根据研究和分析的目的,确定模型的边界,因为任何一个模型都只能反映实际系统的某一部分或某一方面,也就是说,一个模型只是实际系统的有限映象。另一方面,为了使模型具有可信性,必须具备对系统的先验知识及必要的试验数据。特别是,还必须对模型进行形式化处理,以得到计算机仿真所要求的数学描述。模型可信性检验是建模阶段的最后一步,也是必不可少的一步。只有可信的模型才能作为仿真的基础。

仿真建模是仿真执行活动的第一步。其主要任

务是：根据系统的特点和仿真的要求选择合适的算法，当采用该算法建立仿真模型时，其计算的稳定性、计算精度、计算速度应能满足仿真的需要。

第二步是程序设计，即将仿真模型用计算机能执行的程序来描述。程序中还要包括仿真实验的要求，如仿真运行参数、控制参数、输出要求等。早期的仿真往往采用高级语言编程，随着仿真技术的发展，一大批适用不同需要的仿真语言被研制出来，大大减轻了程序设计的工作量。

程序检验一般是不可缺少的。一方面是程序调试，更重要的是要检验所选仿真算法的合理性。这是模型执行的第三步。

第四步是模型运行。有了正确的仿真模型，就可以对模型进行实验，这是实实在在的仿真活动。它根据仿真的目的对模型进行多方面的实验，相应地得到模型的仿真结果。

模型分析活动既是对仿真模型行为数据——仿真输出结果的处理，以便对系统性能作出评价，同时也是对模型的可信性进行检验。输出分析在仿真活动中占有十分重要的地位，特别是对离散事件系统仿真来说，其输出分析甚至决定着仿真的有效性。

在实际仿真时，上述每一个步骤往往需要多次反复和迭代。

仿真技术的应用　仿真技术已经有近100年的发展历史，它不仅用于航天、航空、各种武器系统的研制部门，而且已经广泛应用于电力、交通运输、通信、化工、核能各个领域。特别是随着系统工程与科学的迅速发展，仿真技术已从传统的工程领域扩展到非工程领域，因而在社会经济系统、环境生态系统、能源系统、生物医学系统、教育训练系统也得到了广泛的应用。

在系统的规划、设计、运行、分析及改造的各个阶段，仿真技术都可以发挥重要作用。随着人类所研究的对象规模日益庞大，结构日益复杂，仅仅依靠人的经验及传统技术难以满足越来越高的要求。基于现代计算机及其网络的仿真技术，不但能提高效率、缩短研究开发周期、减少训练时间、不受环境及气候限制，而且对保证安全、节约开支、提高质量尤其具有突出的功效。

总之，仿真技术之所以得到迅速发展，其根本的动力来自应用。仿真技术正是从其广泛的应用中获得了日益强大的生命力，而仿真技术的发展反过来使其得到越来越广泛的应用。

参考文献

肖田元，范文慧. 系统仿真导论. 2版. 北京：清华大学出版社，2010

（肖田元）

jisuanji fangzhen xitong

计算机仿真系统（computer simulation system）　支持仿真活动（系统建模、模型执行、模型分析等，参见**计算机仿真**）的仿真计算机、仿真软件以及仿真环境的总称。

仿真计算机　仿真计算机经历了模拟机、混合计算机、全数字实时仿真机、通用计算机仿真和高性能仿真计算机等阶段。

20世纪50年代模拟机占压倒优势，数字仿真机刚开始发展。50年代末，由于导弹技术发展的需要而诞生混合模拟机及混合计算系统。60年代到70年代中期是混合计算机发展的鼎盛期，与此同时，数字仿真也蓬勃发展，特别是由于完善、价廉的小型通用数字机大量进入市场及数字仿真算法和高级仿真语言研究的发展，在70年代中期，中型混合机的性能价格比已不能与配上完备仿真软件的小型数字机相争。自70年代末期。数字机已逐渐占领仿真计算机的主要市场。80年代以后逐步过渡到全数字计算机。今天的仿真机分布在一个相当宽的型谱与频谱上，它包括个人计算机、工作站、小/中/大型通用数字机、外围阵列机、小巨型机、巨型机、专用仿真工作站、专用数字仿真机、专用混合计算机、并行仿真机等，仿真用户可按需要作出选择。

仿真软件　仿真软件包括仿真支撑软件与仿真应用软件。仿真支撑软件是建立在计算机操作系统之上的以支撑仿真各类活动的软件系统，对计算机而言它是应用软件，但对仿真而言是支撑软件。对基于通用计算机的单机仿真而言，仿真应用软件直接面向计算机**操作系统**，无须仿真支撑软件。但针对特定仿真应用特点定制的计算机硬件体系结构，分布、并行仿真应用的计算过程中往往涉及大量同步、复杂模型解算、不规则的计算与通信模式等，单靠计算机操作系统是难以实现的，需要仿真支撑软件来支持各计算节点协同，典型的如HLA/RTI（参见**高层体系结构HLA**）；有效挖掘仿真并行性（参见**并行仿真**），典型的如GTW（georgia tech time warp，采用乐观同步技术的通用并行离散事件仿真引擎）、SPEEDES（synchronous parallel environment for emulation and discrete-event simulation，基于高性能并行计算平台的同步并行仿真环境），克服分布、并行仿真的开发难度等。

仿真应用软件可分为**仿真语言**和仿真工具包。仿真语言和仿真工具包是基于通用计算机编程语言而又比其更高级的仿真软件系统，为仿真研究人员提供了专门用于建模、仿真实验和仿真结果统计、分析、显示等能力，并最终能自动转换成可执行的代码，使得仿真人员可以不必深入掌握通用计算机高级语言的编程细节和技巧（参见**仿真语言**）。仿真工具包用于某些特定领域的仿真软件，许多已经实现商业化，如多体动力学仿真软件 ADAMS、控制系统仿真软件 MATLAB/SIMULINK，用于有限元分析的 ABQUS、ANSYS 等。

仿真环境　仿真环境是指支持仿真三大活动（系统建模、模型执行、模型分析）（参见**计算机仿真**）的软硬件设施。现代仿真技术，仿真的输入不再满足于键盘、鼠标等常规设备，仿真运行的结果（输出）不再满足于数据、表格、图形等常规形式，而是仿真对象及其环境的三维动画，甚至驱动物理仿真设备；特别是人在回路中的仿真，虚拟现实、沉浸式仿真是构建此类仿真系统的重要组成部分，还要考虑诸如心理活动、行为反应等人类生理特性等。

现代仿真环境一部分依赖于仿真支撑软件与仿真应用软件，还有一些是根据仿真系统的目标专门建立的，它是计算机技术、控制技术、图形图象技术、虚拟现实技术等多种信息技术与其他专业领域技术的综合集成，以达到更加方便开展仿真活动并贴近真实世界的目的。

仿真技术是数值计算、数据处理与知识处理高度融合的技术。以数值计算与数据处理为主的通用计算机难以满足各个领域对仿真技术的要求。未来计算机仿真系统将是数值计算、数据处理与知识处理一体化与智能化系统，其主要特点是：①高速运算的硬软件体系，可进行数值、图形、图像及符号的推理运算；②以信息库为中心的面向数值计算和数据处理的一体化软件系统，具有仿真建模、仿真实验及仿真结果分析的基于信息库管理的集成环境；③高级的人机通信，除数值、文字及符号等信息交互界面外，还具有包括视觉、听觉、触觉、力觉等人类感觉的通信能力；④数值计算、数据处理、知识处理、人的感觉有机地集成。

显然，不断发展的先进的建模/仿真方法学与集成化软件技术、图形技术、并行处理技术、人工智能技术的交叉融合将为新一代计算机仿真系统提供技术支持。

参考文献

1. 李伯虎，王正中. 仿真计算机. 中国计算机用户，1990，6

2. 金伟新. 大型仿真系统. 北京：电子工业出版社，2004

（肖田元）

jisuanji fuzhu chuban

计算机辅助出版（computer aided publishing，CAP）　使用计算机帮助出版从业人员实现提高出版全过程自动化、信息化程度的理论、方法和技术。它综合了**计算机图形学**、**数字图像处理**、文字处理、色彩管理和页面描述语言等多个领域的理论、方法和技术，来构建具有辅助出版功能的系统——桌面出版系统及电子印前系统，以帮助出版从业人员在计算机上完成，从文字、图形、图像和表格等版面元素的输入、编辑、制作，到将这些版面元素排版成为可阅读的页面或版面，直至将这些页面和版面在相应制版、打印或显示设备上进行输出的全部工作。计算机辅助出版能够缩短出版周期；提高出版过程的自动化、信息化水平和出版物的质量；实现许多手工制版无法实现的出版创意效果。

计算机辅助出版起源于 20 世纪 70 年代，直接受到**计算机图形学**、计算机文字处理技术和计算机激光输出设备技术发展的影响和推动。美国施乐公司 Palo Alto 研究中心的两项研究成果，对计算机辅助出版的发展产生了深远影响：其一是 Gary Starkweather 于 1971 年发明的世界上第一台激光打印机 EARS；其二是 John Warnock 和 Martin Newell 于 1978 基于一种面向描述三维图形的 DesignSystem 语言，重新设计实现了一个面向激光打印机的 JAM 语言，该语言后来成为著名的页面描述语言 PostScript 语言的前身。1979 年北京大学计算机研究所，在王选教授领导下，计算机-激光汉字编辑排版系统主体工程技术研制获得成功。在此后数十年的发展过程中，计算机辅助出版经历了从主机到桌面、从黑白到彩色、从批处理到交互、从纸介质到网络等多个前后继承、相互交叉的技术发展阶段，逐步实现了用计算机技术对传统出版产业的自动化、信息化和网络化改造。

所谓出版是指一种包括信息发布、复制、传播过程的社会活动。图书的编辑、印刷、发行过程是一种典型的出版活动。现代出版除基于纸张的印刷出版和办公出版外，更扩展到基于光盘、磁带的音像出版和基于万维网的网络出版，计算机辅助出版可应用于所有这些出版领域。除音像出版较为特殊以外，

其他出版形式的应用场景尽管有很大的不同，但主要技术原理是相同、相近或相通的。这里以印刷出版领域的技术架构为主线进行阐述。

计算机辅助出版解决的问题

计算机辅助出版要解决的问题是，如何在计算机及其外部设备上实现从文图编辑、排版到制版、印刷、发行的印刷出版全过程。基于计算机辅助出版技术的印刷出版工艺流程，划分为版面元素的编辑制作、版面的排版和版面内容的显现及输出这样三个过程。

（1）版面元素的编辑制作　出版物的版面元素种类纷繁，常见的种类包括文本、图片、图形、表格、数学公式、地图、乐谱等。计算机辅助出版通过各种版面元素的专业编辑软件，如图像编辑、图形编辑等软件，来解决版面元素在计算机上的设计制作问题。

（2）版面的排版　所谓排版是指在符合出版工艺规范的前提下，把各种版面元素在版面上进行排列、调整和创意，使之满足设计和美观要求的版面制作过程。在计算机上完成排版过程是通过专门的排版软件来实现的。

（3）版面内容的显现及输出　这里特指将排版结果准确地显现到纸张、显示屏幕、印刷板材等出版物载体的过程。计算机辅助出版涉及的输出设备包括显示器、打印机、激光照排机、直接制版机、直接印刷机等。这些输出设备都可以被理解为某种点阵设备或光栅设备，因此显现及输出问题可以转化为如何将排版结果转换为符合要求的点阵图像问题，即光栅化问题，该问题是通过光栅图像处理器（RIP）来解决的。

计算机辅助出版采用的技术

（1）计算机图形学　计算机辅助出版的主要技术和方法，几乎都以**计算机图形学**为基础。基于图形交互技术、曲线曲面造型技术开发的图形绘制编辑软件，实现图标、花边、插图等图形版面元素在计算机上的直接制作；基于曲线造型技术实现字形设计和字库制作；此外，在排版软件、图片编辑软件、光栅图像处理器中也大量使用了与计算机图形学相关的技术。

（2）数字图像处理　计算机辅助出版的另一个技术基础是**数字图像处理**。在图片编辑软件中，运用图像增强等技术对图片质量进行修正、运用图像分割技术对图片进行自动或半自动的裁剪、运用图像变换技术对图片进行艺术创意和加工；在光栅图像处理器中，基于图像半色调变换技术的挂网技术，实现了文字、图形、图片在点阵设备上的彩色及灰度成像。

（3）文字处理技术　文字是最主要的版面元素之一，计算机辅助出版涉及文字输入、文字编码、文本编辑、文字输出、字库制作等计算机文字处理技术的所有关键领域。在计算机辅助出版中，无论是排版软件还是光栅图像处理器都需要大量不同语言、不同字体的字库提供支持。

（4）排版技术　确定版面元素几何信息和样式信息的专门技术。典型的排版技术有三种，即批处理排版技术、交互排版技术和自动排版技术。批处理排版技术运用置标语言技术将几何信息和样式信息赋予版面元素（这种置标语言被称为排版语言），批处理排版软件的任务是将排版语言形式的排版文件转换为用于输出的页面描述语言形式的排版结果；交互排版软件实际上是一个基于人机交互技术及计算机图形学的版面辅助设计软件，通过光标确定版面元素的几何信息，通过菜单命令及光标确定版面元素的样式信息，并实时显示排版效果，最后输出页面描述语言格式的排版结果；自动排版技术通过将结构化的版面元素存放在数据库中，并辅以排版样式库和规则库，通过自动排版引擎实现无人工干预的自动排版。

（5）页面描述语言　一种面向版面内容显现的计算机语言。该语言在文字、图形和图像等高层语义上，对要显现的版面内容进行描述。典型的页面描述语言如 PostScript、PDF 等。在计算机辅助出版中，页面描述语言是版面的排版过程和版面内容显现及输出过程的纽带，通过页面描述语言实现了这两个过程的松耦合。一个完善的页面描述语言，除了具备丰富的文字、图形和图像的描述能力外，一般还要具备坐标空间变换、裁剪、颜色空间转换的描述能力，以及与设备无关的特性。国际标准化组织 ISO 在已有页面描述语言的基础上，制定了标准页面描述语言 SPDL（ISO/IEC DIS 10180）。

（6）色彩管理技术　计算机辅助出版应用环境是一个彩色设备聚集的环境。各种彩色设备的色域范围各不相同，呈现为相互交叉又不完全覆盖的局面。色彩管理技术的目标是，基于色度学理论，实现色彩信号在不同色域设备间能够相互转换，且转换前后的颜色外貌要尽可能地相似。具有色彩管理功能的、设备独立的、开放的、标准的色彩处理系统称为色彩管理系统。实现色彩管理目标的基本方法是，选用一个设备独立的标准色彩空间，如 CIE Lab

彩空间，作为色彩转换的中介空间和精度评价空间，通过为各种设备建立色彩特性文件，如 ICC Profile 文件，构建起设备色彩空间与该标准色彩空间的转换关系，在此基础上，不同彩色设备之间的颜色信号就可以通过标准色彩空间作为桥梁间接完成相互间的转换。色彩管理技术已广泛应用于图形、图像编辑软件和排版软件，并成为光栅图像处理器的核心功能。

（7）光栅图像处理器技术　将排版结果光栅化的专门技术。运用该技术实现的光栅图像处理器，是计算机辅助出版系统的核心。由于在页面描述语言形式的排版结果中，文字、图形、图像三类版面元素的数据形态差别很大，因此对这三类版面元素存在不同的光栅化技术。光栅图像处理器技术主要包括如下几部分：

① 对页面描述语言的解释执行　典型的是对 PostScript 文件进行语法分析，并对分析结果进行相应的解释执行。

② 对图形对象的光栅化处理　运用计算机图形学的理论和方法，进行曲线、曲面的绘制及填充。

③ 对文字对象的光栅化处理　运用文字处理技术，根据指令及文字的编码，从字库中查找出各个字符的轮廓曲线数据进行填充。

④ 对图像对象的光栅化处理　运用数字图像处理中半色调变换技术，将图像转变为二值点阵数据，该技术也称为挂网技术。彩色、灰度的文字和图形也需要进行挂网处理。

⑤ 色彩空间转换处理　运用色彩管理技术，将排版结果中用到的各种色彩空间的数据，都转换成显现设备使用的色彩空间的数据。最典型的转换是从显示、扫描所使用的红绿蓝（RGB）颜色空间向印刷、打印所使用的黄品青黑（CMYK）颜色空间转换，这种转换技术也称作为分色技术。

随着网络出版技术的发展，针对网络出版特点的页面描述语言、页面设计制作方法和技术得到了迅速发展。移动阅读设备，如手机、电子书、平板计算机的兴起为计算机辅助出版提出了许多新的要解决的问题，例如，移动阅读设备屏幕的分辨率千差万别，如何实现内容制作一次，多分辨率自适应展现；如何让印刷出版和网络出版的内容在小屏幕上自适应展现；移动阅读设备的用户，对内容的需求更加个性化，如何自动推荐和制作符合不同口味用户的个性化出版物。这些问题都是计算机辅助出版继续发展的动力来源。

参考文献

1. Homann J-P. Digital color management: principles and strategies for the standardized print production(X. media. publishing), Springer, 2010

2. Surhone L M, Timpledon M T, Marseken S. F. page description language. Betascript Publishing, 2010

3. Marin J. Digital prepress primer, Printing Industries Press, 2006

4. 钟云飞，唐少炎. 计算机排版原理. 北京：印刷工业出版社，2005

（陈晓鸥）

jisuanji fuzhu cidian bianzuan

计算机辅助词典编纂（computer-aided lexicography）　基于大规模语料库以人机互助方式进行词典编纂的过程、技术和方法。目标在于通过**自然语言处理**技术和语料库技术的介入提高传统词典编纂的效率和质量，革新词典编纂的手段。由于倡导使用语料库以及强调语料库在词典编纂中的价值，所以一般也称为**基于语料库的词典编纂**（corpus-based lexicography）。

最早的计算机辅助词典编纂工作可以追溯到 20 世纪 60 年代，不过真正成功的案例出现是在 20 世纪 80 年代。历经发展，计算机辅助词典编纂目前已成为发达国家词典编纂实践中的基本手段。随着语料库规模的增长和语料库技术的演进，基于语料库的词典编纂技术也在不断深入和发展。

计算机辅助词典编纂的基本理念与语料库语言学的基础理念一脉相承，即词典编纂和语言研究应以语言应用而非语言能力为中心、应以语言描写而非语言直觉为中心、应以语言的定量及定性模型为中心、应以经验主义而不是理性主义的研究方法为中心。由于语料库是语言实际使用的样本，只有立足于语料库进行词典编纂，才能保证词典的编写有客观依据，才能保证词典的编写不与语言的真实使用脱节。与传统词典编纂不同，基于语料库的词典编纂一方面通过大量使用源自语料库的频率信息、搭配信息和例证信息提高了词典编纂的质量，另一方面借助信息技术提高了词典编纂的效率。

计算机辅助词典编纂需要进行两个方面的基础研究和建设：①服务于词典编纂的语料库。语料库建设需要考虑其构成的代表性，经过适当标注和加工。服务于词典编纂的语料库一般规模较大，且具有开放性，能监控语言在一段时期内的变化，属于所

谓的监控语料库。②服务于词典编纂的计算机软件系统。该系统要为词典编纂人员编写词典提供一个友好的格式化界面，同时提供词典编纂任务的流程管理，尤其重要的是该系统需要提供基于语料库的词汇知识分析功能。可以根据词典编写人员的要求，提供从语料库中检索每个词语的相关句列（concordance）功能以及带语境的关键词居中（key word in context，KWIC）显示界面、提供词频信息和基于语料库的搭配统计信息等，将语料库中的隐式词汇信息显式提供给词典编纂人员。

国际上，服务于英语词典编纂的工作较为突出。目前为英语词典编纂提供服务的语料库有COBUILD语料库、朗文语料库网、BNC语料库等，基于语料库成功编纂的英语辞书也为数不少，例如《柯林斯COBUILD高阶英语学习词典》、《朗文当代英语词典》、《牛津高阶学习词典》等。

参考文献

1. Landau S I. 词典编纂的艺术与技巧. 章宜化，夏立新，译. 北京：商务印书馆，2005

2. Sinclair J. Corpus concordance and collocation. Oxford：Oxford University Press，1991

（常宝宝）

jisuanji fuzhu fanyi

计算机辅助翻译（computer-aided translation，CAT）　基于人机互助策略进行自然语言翻译的过程、技术和方法，又称机器辅助翻译（machine-aided translation）。目标不是利用全自动翻译取代人工翻译，而是以人机分工合作方式提升翻译从业人员的工作效率和翻译质量。它从理念上与目标是实现全自动翻译的**机器翻译**有显著不同。

计算机辅助翻译技术在学术界以及产业界受到关注和重视的原因在于，目前缺乏全自动高质量的机器翻译技术，全自动机器翻译技术很难被专业翻译人员所接受。

根据翻译人员人工介入程度的不同，计算机辅助翻译技术可分为两种不同的类型：① 人助机译（HAMT）技术，其中机器是翻译的主体完成者，在机器翻译的基础上加入译前编辑（pre-editing）和译后编辑（post-editing）之类的人工干预，这种人工干预或在机器翻译之前人工化解机器翻译尚不能有效应对的复杂翻译现象，或在机器翻译后人工改进机器译文质量，从而达到生成高质量译文的目标。② 机助人译（MAHT）技术，其中翻译主要由人工完成，机器在翻译过程中提供翻译记忆、术语翻译等方面的支持作用。从目前的发展情况来看，机助人译技术较好地适合了翻译从业人员的需要，较人助机译技术更为成功。基于狭义理解，计算机辅助翻译有时也仅指机助人译技术。

多数计算机辅助翻译系统立足于为翻译人员提供一个集成化翻译环境，将多种对人工翻译有益的计算机技术集成提供给翻译人员使用。这种集成翻译环境一般称为翻译人员工作站（translator's workstation）。在众多辅助翻译技术中有两种技术尤其获得翻译从业人员的认可和使用，分别是：①翻译记忆（translation memory）技术，这种技术将翻译人员的历史翻译记录下来或将现存翻译对齐记录在一个称为翻译记忆库的数据库中，翻译人员在接到新的翻译任务时，辅助翻译系统以某种精确或模糊匹配的策略在翻译记忆库中寻找已有或可资借鉴的翻译结果，从而达到翻译复用（translation reuse）以提升翻译效率的目标。②术语管理和预翻译技术，这种技术的核心是维护一个双语（或多语）对照的专业术语库，术语库中的内容可由翻译人员人工建立，也可通过术语自动发现和提取方式从专业语料库中提取，在翻译人员接到专业翻译任务时，专业术语库可以用来对翻译任务中的术语进行识别和预翻译，这可以有效应对翻译人员因不具备专业知识对相关术语不能正确翻译的问题以及减缓由于多人合作引起的术语翻译不一致现象。计算机辅助翻译系统的常见功能还包括文件格式处理和翻译项目管理等。

在当今全球化和国际化时代，各行各业产生了大量本地化翻译任务，需要将各种产品和服务针对不同文化和语言进行适应性调整，本地化翻译任务的特点是工作量大、专业性强、时间紧迫，但翻译任务多有重复，在这种环境下，以翻译记忆和术语辅助翻译为核心的技术大有用武之地，极大地提高了翻译效率和质量。

参考文献

1. Bowker L. Computer-aided translation technology：a practical introduction. Ottawa：University of Ottawa Press，2002

2. Austermühl F. Electronic tools for translators. Manchester：St. Jerome Publishing，2001

（常宝宝　刘群）

jisuanji fuzhu gongcheng

计算机辅助工程（computer aided engineer-

ing,CAE) 采用计算机辅助手段对复杂工程和产品的力学性能分析计算、结构性能优化设计等问题求解的一种近似数值分析方法。计算机辅助工程(CAE)技术可用于工程和产品设计中的各种分析计算与分析仿真,包括工程数值分析、结构与过程优化仿真、强度与寿命评估、运动/动力学仿真等,可以确保工程和产品设计的合理性,减少设计成本。例如,CAE 采用基于仿真的优化设计,可找出最佳设计方案,降低材料消耗和成本;可在产品制造或工程施工前预先发现潜在问题;可模拟各种试验方案,减少试验时间和成本。基于高端 CAE 技术的虚拟样机,在很大程度上替代了传统设计中的"物理样机验证设计"过程,能够预测产品在整个生命周期内的可靠性。

早在 20 世纪 50 年代末、60 年代初,国际上就开始了计算机辅助工程(CAE)技术的研究,特别研发了具有强大功能的有限元分析程序。其中,最为著名的是美国国家航空航天局(NASA)为满足当时航空航天工业对结构分析的迫切需求,委托美国计算科学公司和贝尔航空系统公司于 1969 年开发的第一个大型有限元应用程序 NASTRAN 系统。此后,德国的 ASKA、英国的 PAFEC、法国的 SYSTUS、美国的 ABQUS、BERSAFE、BOSOR、COSMOS、ELAS、MARC 和 STARDYNE 等公司的 CAE 软件产品层出不穷。美国于 1998 年成立工程计算机建模与仿真学会,致力于 CAE 技术的研究和工业化应用,极大推进了 CAE 技术和系统的发展。随后,其他国家也纷纷建立了类似的学术组织。目前,在国际上具有影响力的 CAE 软件产品有:NASTRAN、ANSYS、ADINA、Moldflow、I-DEAS、Hyperworks、Abaqus、LS-DYNA、Pam Crash、AutoForm、Madymo 等。

经过 50 多年的发展,计算机辅助工程(CAE)理论和算法经历了从蓬勃发展到日趋成熟的过程。从 CAE 技术发展来看,主要有三种技术:①有限元法(finite element method);②边界元法(boundary element method);③差分法(finite difference method)。其中,有限元法的应用最为广泛。CAE 分析的核心思想是结构的离散化,即将实际结构离散为有限数目的规则单元的组合体;实际结构的物理性能可以通过对离散体进行分析,得出满足工程精度的近似结果来替代对实际结构的分析,这样可以解决很多实际工程需要解决而理论分析又无法解决的复杂问题。CAE 分析的基本过程是将一个形状复杂的连续体的求解区域分解为有限的形状简单的子区域,从而将一个连续体简化为由有限个单元组合的等效组合体;通过连续体的离散化将求解连续体的场变量(应力、位移、压力和温度等)问题简化为求解有限的单元节点上的场变量值,得到一个代数方程组;求解后得到近似的数值解,其精度取决于所采用的单元类型、数量以及对单元的插值函数。

利用 CAE 软件对工程或产品进行性能分析和模拟,通常分为下面三个步骤:

(1) 前处理 采用**计算机辅助设计**(CAD)等技术对工程或产品建立模型,进行网格剖分,形成合理的有限元分析模型。

(2) 有限元分析 对有限元模型进行单元特性分析、有限元单元组装、有限元系统求解和有限元结果生成。

(3) 后处理 根据工程或产品模型与设计要求,对有限元分析结果进行用户所要求的加工、检查,并以图形方式(如显示位移、温度、压力分布的等值线图、彩色明暗图、动态显示图等)提供给用户,辅助用户判定计算结果与设计方案的合理性。

CAE 技术被广泛应用于不同行业和不同类型的产品研发中,已成为航空、航天、机械、土木等领域工程和产品结构分析和结构优化的重要工具。同时,CAE 系统也是计算机辅助 4C 系统(计算机辅助设计(CAD)/CAE/**计算机辅助工艺规划**(CAPP)/**计算机辅助制造**(CAM))的一个重要环节。

计算机辅助工程 CAE 技术的主要发展趋势体现在以下几个方面:

(1) CAE 系统的核心分析功能进一步增强,能够实现可变形体系与多体耦合分析、多相多态介质耦合分析、多物理场耦合分析、多尺度耦合分析、结构与构件及其材料的一体化设计计算与模拟仿真等;在分析问题规模方面,能够求解上千万阶方程组。

(2) CAE 系统相关功能进一步强大,其三维实体建模、复杂的静态、动态物理场的**虚拟现实**技术将有很大发展,能够实现对复杂工程、产品的实时和真三维仿真。

(3) CAE 系统的性能进一步提升,出现一些基于新计算架构的 CAE 系统,如基于超级计算机和计算机群的并行计算 CAE 系统、基于网格计算的 CAE 系统等。

(4) 集成化 CAE 系统能够与 CAD 系统实现双向无缝集成,基于**产品数据管理**(PDM)(**产品生命周期管理**(PLM))实现 CAD /CAE /CAPP /CAM

集成系统。

(5) 专业化　在通用 CAE 软件平台上开发专业化的应用软件，建立企业级的 CAE 分析技术标准化软件，简化分析方法，提高 CAE 应用效益，以此来建立和提升企业开发和研制的能力。

参考文献

崔俊芝. 计算机辅助工程(CAE)的现状和未来. 中国科学院院士报告，2006

（林兰芬）

jisuanji fuzhu gongyi guihua

计算机辅助工艺规划(computer aided process planning，CAPP)　利用计算机设计产品零件机械加工工艺规程的技术。机械加工工艺规程设计所要解决的主要问题是根据零件几何形状、尺寸、加工精度、表面粗糙度和热处理等技术要求，确定零件的加工方法、加工顺序、所需的机床、刀具、夹具、量具和切削参数等。计算机辅助工艺规划(CAPP)综合应用**计算机图形学**、**工程数据库**和**人工智能**等计算机技术解决这一复杂的多目标规划问题，不仅克服了手工设计工艺规程存在的效率低、一致性差、难以积累并充分利用工艺专家的经验和知识等缺点，而且提高了工艺规程的质量和标准化、规范化程度，并且可对工艺技术文件进行统一的管理，达到缩短产品生产准备周期、降低生产成本的目的。计算机辅助工艺规划(CAPP)是连接**计算机辅助设计**(CAD)和**计算机辅助制造**(CAM)的桥梁，是计算机集成制造系统的重要组成部分，也是企业实现产品设计制造数字化和一体化的一个关键环节。

计算机辅助工艺规划(CAPP)的研究可追溯到20世纪60年代末，Niebel于1965年首次提出CAPP思想。CAPP发展史上的一个重要里程碑是美国J. Jiokoff等人于1976年开发的自动工艺规划系统(CAMI，automated process planning system)。此后，CAPP技术领域的研究得到了快速发展，其间经历了检索式、派生式、创成式、混合式、专家系统、工具系统等不同的发展阶段，并涌现了一大批CAPP原型系统和商品化的CAPP系统。

根据工作原理不同，CAPP系统可分为检索式、派生式、创成式、工具化CAPP系统等。

(1) 检索式CAPP系统　将设计好的零件标准工艺进行编号，存储在计算机中。当制定新零件的工艺规程时，可根据输入的零件信息进行搜索，查找合适的标准工艺。

(2) 派生式CAPP系统　其基本原理是利用零件的相似性，即相似零件有相似的工艺规程。新零件的工艺规程是通过检索相似零件的工艺规程，加以自动筛选或编辑而成。根据零件信息的描述与输入方法不同，派生式CAPP系统又分为基于成组技术(GT)与基于特征的派生式CAPP系统。

(3) 创成式CAPP系统　根据输入的零件信息，依靠系统中的工程数据和决策方法自动生成零件的工艺过程。

(4) 工具化CAPP系统　以解决工艺设计中的事务性、管理性工作为首要目标，优先解决工艺设计中工艺资料查找、卡片填写、数据计算、分类汇总与数据集成等烦琐重复而适合使用计算机辅助方法的问题，强调提升制造企业的工艺规划的效率、标准化和集成性。

CAPP技术已广泛应用于不同制造业行业企业的产品研发和设计中，已成为航空、船舶、汽车、航天、装备等行业产品工艺设计与管理的重要工具，是计算机辅助4C系统(CAD/CAE/CAPP/CAM)的一个重要环节，是实现面向全生命周期的产品数字化设计与制造集成的关键。

计算机辅助工艺规划(CAPP)的技术主要发展趋势为：①工艺设计的可视化。工艺设计模式从二维向三维转变，CAPP可为工艺设计人员提供一个接近真实的可视化工艺设计环境，使工艺设计人员能够进行包括建立和提取三维模型中的加工特征信息、加工过程仿真和校验、装配过程仿真优化、加工资源规划等高层次的工艺设计工作。②工艺设计的智能化。对工艺专家的经验和知识进行表达、获取和应用，可以根据三维产品模型包含的丰富信息进行创新型的工艺设计。③工艺设计的集成化和协同化。实现CAPP系统与CAD、CAM、**产品数据管理**(PDM)/**产品生命周期管理**(PLM)、制造执行系统(MES)、**企业资源计划**(ERP)等系统的紧密集成，实现企业之间、企业与客户之间、设计人员之间的协同工艺设计。④工艺设计的专业化。在建立通用开放CAPP平台的基础上，面向不同行业、不同企业的需求，根据行业的工艺设计的经验和特点(如产品特点、工艺装备特点、工艺设计过程特点、工艺知识累积特点等)，定制开发行业专用的CAPP系统。

从系统角度看，未来CAPP系统将在应用范围、深度和水平等方面进行拓展，主要发展趋势表现为：①面向产品生命周期的CAPP系统。CAPP与**产品生命周期管理**(PLM)的集成是制造业企业应用集成的关键之一，支持产品生命周期的CAPP系统将

日显重要。②基于知识的 CAPP 系统。在解决工艺设计效率、标准化规范化问题的基础上，有效地总结、沉淀企业的工艺设计知识，实现基于知识的创新工艺设计。③基于三维产品模型的 CAPP 系统。在可视化的工艺设计环境中，利用三维产品模型进行三维工艺设计，包括三维轻量化产品建模、三维工艺建模、基于三维产品模型的加工工艺设计、装配工艺规划与仿真分析、工艺知识管理等。④基于平台技术、可重构式 CAPP 系统。具有二次开发功能与可重构能力，支持个性化工艺设计和工艺设计的变化，能够持续满足企业不断变化的专门化与个性化需求。

参考文献

1. 邵新宇，蔡力钢. 现代 CAPP 技术与应用. 北京：机械工业出版社，2004

2. 张振明，许建新，贾晓亮，等. 现代 CAPP 技术与应用. 西安：西北工业大学出版社，2004

（林兰芬）

jisuanji fuzhu jiaoxue xitong

计算机辅助教学系统（computer-assisted instruction system）　为提高教学质量与教学效率和弥补传统教学模式的不足，利用计算机和网络辅助进行教学活动的**应用软件系统**。

自计算机问世以来，计算机在教育领域中的应用一直受到人们的关注。其中特别受到重视的是计算机辅助教学（computer-assisted instruction，CAI）和计算机管理教学（computer managed instruction，CMI）。CAI 综合应用**计算机网络**、**人工智能**、知识处理和多媒体等技术，它既可作为常规课堂教学的补充手段，也可以部分地替代教师进行课程的教学。CMI 指的是使用计算机进行学生注册、选课、排课、教学评估、学分统计、成绩记录与查询等管理。CAI 和 CMI 是计算机辅助教育（computer based education，CBE）的两个主要组成部分。

与传统教学手段相比，对于那些用语言文字难以表达的抽象内容、动态的变化过程和复杂的微观结构等，CAI 通过使用图表、动画、声音等多媒体技术进行演示或模拟其变化过程，可显著改善教学效果。采用 CAI 技术的训练课和习题课，以学生为中心，按各自的进度进行，适合个性化教学，能充分发挥学生的学习主动性。CAI 还利用计算机的交互特性实现双向教学，克服传统教学过程中学生只是单向接受知识的被动局面。

CAI 系统除了必要的硬件和系统软件、支撑软件之外，**课件**（courseware）是其最重要的一个组成部分。课件是按教学目标进行设计并反映某种教学策略的，用于传递教学内容、实施课程教学活动的一种计算机应用软件，因而是 CAI 系统的核心。

从教学角度分析，课件由教学内容、教学策略和学生模型三个部分组成。不同的教学内容和教学目标应采用不同的教学策略，不同的教学策略适合于不同认知能力和学习要求的学生。

从软件角度分析，课件包含数据（与教学内容相关的教材、讲稿、演示、习题、参考资料等）、控制（用于对学习过程进行引导、处理、诊断和评价，决定教学过程如何进行）及人-机接口（确定学习者与课件的交互方式以提高教学效果）三个部分。通常，课件有如下一些类型：①示教型：用于演示抽象、复杂的现象和过程，适合课堂教学使用。②实验模拟型：模拟电子电路实验、化学实验等，可节省实验费用。③授课型：适用于自学某些课程，对职业培训和成人教育较为合适。④训练测试型：可提高外语、医学、程序设计等课程的课外学习效率。⑤游戏型：通过游戏达到预定的教学目标，把趣味性、教育性、科学性融为一体，做到寓教于乐。⑥咨询型：学习者主动提出问题，由课件给予回答和说明，有利于学生自主学习。此外还有问题求解型、发现学习型等。

课件的开发过程包括：教学需求和教学目标的论证，课程的教学设计，课程的结构设计，课件的制作、测试和评价，运行维护等。一个优秀课件的开发往往需要教育专家、任课教师和软件工程师等几方面人员的参与，并在开发和使用过程中反复听取学生的意见。课件开发中，有一类称作著作工具（authoring tool）的软件工具，其主要特点是简单易用，不要求或者很少要求用户具有编程语言、数据结构和算法设计等方面的知识，用户经过短期学习之后就可进行课件开发。

目前 Internet 上已经有大量的课件资源，其中最有名的是麻省理工学院的开放课件 OCW（MIT open course ware）。OCW 是一组免费的课件资源，截至 2010 年底已有约 2000 门本科和研究生课程的课件可供世界各地使用。开放课件的出现开创了互联网时代共享教育资源的理念和行动，得到了许多国家和地区教育机构和有关专家的响应。

计算机在教学方面的应用早在 20 世纪 60 年代就已经开始了，美国伊利诺伊大学开发的 PLATO 系

统就是最早的计算机辅助教学系统之一。自 80 年代开始,随着微型计算机在大、中、小学的普及使用,计算机辅助教学从大型机转向桌面系统。这一时期,教师积极参与课件开发,专业的教育软件公司也相继出现,计算机辅助教学系统得到了蓬勃发展。

进入 21 世纪之后,随着 Internet 的普及和 Web 技术的广泛应用,计算机辅助教学系统进入了一个新的发展阶段。例如,实时(利用视频会议系统、虚拟教室)或非实时(如利用网络课件、视频点播)向学生讲授课程;实时(如利用在线交谈)或非实时(如利用电子邮件)对学生进行个别辅导;学生与学生、学生与教师利用 BBS、**电子邮件**、QQ、博客、维基、播客等手段实时或非实时地进行课外讨论;各种教学管理活动也全面采用了 Web 技术,包括网上注册、网上选课、作业递交、网上查分等。

随着 CAI 深度和广度的发展,国内外学者开始更多地使用 WBI(Web based instruction)、WBL(Web based learning)、WBT(Web based training)、WBE(Web based education)、online-learning(网络学习)、E-learning(数字化学习)等概念和名称,其中尤以网络学习和数字化学习比较流行。它们虽各有不同侧重,但都是强调通过 Internet 和数字化技术为学习者提供一个具有全新沟通机制和丰富教学资源的有效学习环境,以开展多种类型的教育教学活动。更进一步的目标则是试图和追求实现一种全新的学习方式,以充分体现学习者的主体地位和协作学习的作用,根本改变传统的教学结构和教育的本质。

参考文献

1. http:// en. wikipedia. org/wiki/Courseware # Courseware, 2011.2

2. http:// en. wikipedia. org/wiki/MIT _ OpenCourseWare, 2011.2

(张福炎)

jisuanji fuzhu ruanjian gongcheng

计算机辅助软件工程(computer aided software engineering,CASE) 参见**软件工程**。

jisuanji fuzhu sheji

计算机辅助设计(computer aided design,CAD) 利用计算机帮助设计人员完成设计任务的理论、方法和技术。它是综合**计算机图形学**、**人机交互技术**、分析优化技术和设计方法学等多个领域的技术,建立的具有辅助设计功能的系统。在 CAD 系统帮助下,设计人员能够进行设计方案的生成、分析、计算、优化、综合、修改及文件编制工作,并通过提供遵循业界标准的数据接口实现与计算机辅助工程(CAE)及**计算机辅助制造**(CAM)系统的数据交换。专用 CAD 系统通常还含有领域知识,辅助设计人员进行优化设计和计算。CAD 可以减轻设计人员的劳动,缩短设计周期,提高设计质量。**计算机辅助工程**(CAE)、**计算机辅助优化设计**(CAO)、**电子设计自动化**(EDA)等都是与 CAD 密切相关的技术领域。

计算机辅助设计是伴随着计算机图形学和计算机辅助制造技术发展起来的。20 世纪 50 年代初,美国麻省理工学院伺服机构实验室用 Whirlwind 计算机开发了第一台自动控制铣床。1958 年, S. Coons 提出了计算机辅助设计这一概念。1962 年, I. E. Sutherland 在麻省理工学院开发的 Sketchpad 人机通信的图形系统标志着计算机图形学的产生,其方便、直观的交互方式和图形显示功能使计算机辅助设计得到了迅速发展。70 年代,完整的 CAD 系统开始形成并逐步商品化,曲面造型系统 CATIA 为人类带来了第一次 CAD 技术革命。80 年代,图形工作站成为 CAD 典型环境,计算机辅助设计方法从绘图进入到参数化造型阶段,并开始向标准化、集成化、智能化方向发展,产生了 AutoCAD、Pro/E、Protel 等各种设计软件。90 年代,CAD 硬件支撑从专用的图形工作站扩展到个人计算机,设计的范围也从单纯的形状、结构等的设计扩展到性能的分析、模拟和设计优化。21 世纪以来,CAD 软硬件支撑平台都建立在互联网环境下,CAD 技术更加强调基于网络的协同集成设计技术。

计算机辅助设计的对象纷繁复杂,涉及的范围比较广泛,从需要满足复杂工程需求的机械产品设计(包括一般机械产品设计和汽车、造船、航空、航天等复杂产品设计)、电子产品设计、建筑设计到追求创意和美感的美术设计、广告设计、时装设计,其中应用的专业知识、设计方法、功能需求均不相同。对于这些不同的设计领域,CAD 系统的结构、组成、功能均存在很大差异,但系统实现的一般性原则、原理和所采用的计算机技术却是共同的。

机械、电子、建筑是 CAD 传统的应用领域,开发技术和应用均已取得很大成功。这里主要以计算机辅助机械设计为例加以阐述。这不仅因为 CAD 概念首先产生于机械产品设计领域,而且在机械工程领域,计算机辅助设计已形成了一些成熟的理论、技

术和产品，并得到了成功的应用。

计算机辅助设计解决的问题

以机械产品设计为例，根据设计的各阶段工作，CAD 要解决以下几方面问题：

(1) 造型　即建立设计模型。造型的主要工作是建立产品的几何形状，输入产品的设计属性，如物理特性、材料特性、尺寸、公差等。造型的两个技术要素是给用户提供设计手段和建立产品模型的表达机制。设计手段是用户用以建立设计模型的方式、方法，如特征化、参数化技术（参见**几何造型**）；表达机制是设计模型的表示方法，如自由曲面表达、实体表达。设计手段和表达机制的不同导致了不同类型的造型技术，如曲面造型、实体造型、特征造型等。先进的造型系统要求将曲面、实体、特征等多种技术融于一体，以便在计算机上建立复杂的设计模型。

(2) 分析仿真　采用计算机辅助分析（CAE）技术实现对设计对象的分析模拟功能，如热力、静力、动力分析，动态装配、分离等运动仿真等，并采用可视化手段对分析结果进行展示。目前这类分析主要采用有限元方法实现。

(3) 优化　根据仿真分析的结果，优化设计模型，得到满足设计要求的最佳设计结果。建模、分析、优化的过程往往需要多次循环，也要用到**计算机辅助优化设计** CAO 技术。除了专用的优化设计软件外，有些通用的优化系统（如 Isight 等）综合了多种优化算法，能对系统中的多个参数进行迭代优化。

(4) 数据交换　包括 CAD 系统之间的数据交换以及 CAD 系统与 CAE、**计算机辅助工艺规划**（CAPP）、CAM 等系统间的数据交换。不同系统间的数据交换主要解决特征定义不一致问题，如从 CAD 模型转到 CAPP 需建立设计特征与加工特征之间的映射。数据交换可以通过符合标准的中性文件进行，也可通过建立集成化产品模型进行。

计算机辅助设计采用的技术

(1) **计算机图形学**　是计算机辅助设计中采用的重要技术。它主要包括造型、图形显示、图形标准等内容。产品几何形状的建立、表达、显示等均需用计算机图形学实现。造型技术主要解决产品几何形状的表达机制和构造方法；图形显示技术是根据产品的几何形状表达在屏幕上显示该产品的形状（参见**真实感图形生成**）；图形标准主要解决 CAD 系统的易移植性及设计数据的可交换性。

(2) **人机交互技术**　为计算机辅助设计提供图示化用户界面和交互数据输入机制。CAD 系统中设计模型的建立、修改等工作需要用户进行大量操作，系统需要不断地接收用户的输入事件并迅速做出反应，交互性很强。人机交互技术为用户提供方便灵活的特征修改、删除及操作回退等功能，在 CAD 系统中必不可少。

(3) 数据管理技术　包括设计过程中设计数据管理和维护以及设计完成后设计结果的管理。产品设计过程中涉及大量的几何、特征及操作数据，其结构复杂，联系众多，需要设计合理的结构和机制以维护各种对象之间的依赖关系，以保证操作后模型的正确性。设计结果的管理则主要是为了满足设计的重用及产品生产过程中对设计数据的调用。**产品数据管理**（PDM）为解决这一问题提供了方案。

(4) 分析优化技术　机械和建筑设计中的计算机辅助分析主要采用**有限元方法**，体网格的划分是决定有限元分析的精度和速度的主要因素。近年来，无网格技术在计算机辅助分析中的应用逐步增多。

计算机辅助设计的发展趋势

目前，CAD 已广泛应用于电子、建筑、机械、航空航天、汽车、造船等众多的工程领域，并取得了巨大的经济效益。计算机辅助设计 CAD 未来的发展主要方向是：

(1) 基于 Internet 的协同设计　基于万维网的计算机辅助协同设计系统将能够通过因特网对远程的模型进行异地设计，从而能使异地设计人员方便地交流设计思想，并能在外协件的装配等方面尽早地发现冲突，缩短设计周期。

(2) 概念设计　广义上的概念设计包含了从产品的需求分析到进行详细设计之前的设计过程。它包括功能设计、原理设计、形状设计、布局设计和初步的结构设计。通过知识库和推理帮助设计人员完成概念设计，突破传统 CAD 系统只能进行辅助建模和辅助分析等的局限，真正赋予 CAD 系统辅助设计的功能，是 CAD 新的发展方向。

(3) 多领域集成建模和优化　一个复杂产品往往涉及机、电、液、控、热等多个学科，不同领域间的设计参数存在着耦合，对某一领域最优的设计很可能以降低另一领域的性能为代价。进行多领域统一建模，理清领域间设计参数的关联关系，通过多学科协同优化实现领域间的平衡从而达到系统的整体最优，是 CAD 的难点和热点。基于多学科协同优化的**计算机辅助优化设计** CAO 技术是这方面的主要发展方向。

（4）虚拟样机技术　通过对产品的数字化建模和全方位仿真在设计阶段实现产品的缺陷检测和性能分析。其主要技术基础还是计算机辅助设计中的分析、显示等技术，与传统的计算机辅助设计和分析相比，它更强调整体，强调多领域的协同，理想状态应是对产品全生命周期的全方位的测试和仿真。它可以看作是传统 CAD/CAE 技术的发展和延伸。

参考文献

1. 潘云鹤. CAD 系统与方法. 杭州：浙江大学出版社，1996

2. 孙家广，等. 计算机辅助设计技术基础. 北京：清华大学出版社，2000

3. 谭建荣，等. CAD 方法与技术. 北京：科学出版社，2005

（童若锋）

jisuanji fuzhu youhua sheji

计算机辅助优化设计（computer aided optimization design，CAO）　将最优化技术与**计算机辅助设计**技术结合的一种现代设计方法。CAO 根据设计的理论、方法和标准规范，以计算机支持设计对象和目标的建模，通过最优化方法进行优化计算，辅助设计人员得到最优设计方案和结果。CAO 是集设计思考、绘图、计算、实验于一体的现代、科学的设计方法。

以前，工程设计人员根据自己在设计实践中的经验产生了许多优化方法，诸如图解法、黄金分割法等。但由于优化计算一般需要进行反复迭代，很难采用手工计算的方式完成。随着计算机技术的发展，将最优化技术引入设计领域进行辅助设计逐步发展起来。以前需要设计者将优化问题转化成为解析形式的数学公式，然后进行优化计算。随着各种优化技术和计算手段的不断发展，设计者开始使用诸如仿真模型、试验设计等技术手段开展优化设计。

计算机辅助优化设计 CAO 其优化内容及其过程可用图 1 表示。图中，F 是目标函数，C 是约束函数，x 为优化变量。

（1）建立对象模型　建模环境是计算机辅助优化的重要部分。对象模型的描述形式可以是数学模型，也可以是计算机可实现的一般模型（比如仿真模型、表格模型、图形模型等）。

（2）建立优化模型　优化模型是优化目标的描述，多采用目标函数与约束函数形式。目标函数根据用户要求的优化目标，以函数形式定义优化变量、优化变量与对象模型的关系、优化变量间的关系等；

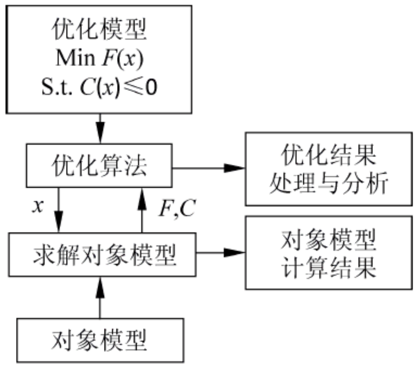


图 1　计算机辅助优化

约束函数以数学等式或不等式形式定义优化变量、对象模型变量（参数）的约束条件。在计算目标函数极值时必须以满足约束函数为前提。

（3）确定优化算法　这是 CAO 的核心内容。根据优化模型的类型，选择或开发合适的优化算法，编制或调用优化计算程序。目前的优化算法较多，但都有其适用范围。要针对研究对象的性质和优化目标函数的特征，重点考察算法的优化能力及优化效率，选择适当的优化算法。

（4）优化计算　基于 CAO 支撑环境进行目标函数优化。在优化过程中，先要由用户确定优化的初始点；再通过优化算法确定优化变量值并以优化变量值替代对象模型相关变量（或参数）的值，然后计算对象模型的输出，进而计算目标函数值。如果对象模型比较复杂，或者要采用仿真的方法来计算，则须选择恰当的仿真算法，以保证对象模型计算的准确性并尽可能减少仿真计算时间。在优化计算过程中，CAO 支撑环境往往提供人机交互界面，显示优化过程及相关变量和目标函数值，使用户实时了解优化过程，对优化过程进行控制。

（5）优化结果处理与分析　CAO 系统对优化的结果进行处理与分析，辅助用户做出正确的决策。

CAO 技术已广泛应用于机械、电子、建筑等传统设计领域，对各种现代复杂装备向高、精、尖方向的发展也起到了促进作用。CAO 广泛应用反过来也有力地促进了 CAO 本身的发展，例如，多学科设计优化（MDO）技术就产生于航空领域的优化设计。目前，CAO 技术也正不断被应用于工业工程、经济管理乃至公共管理等非传统设计领域，比较典型的应用是物流系统的优化设计、投资体系的优化设计等。从理论上说，大部分能够以参数形式建立数学模型的系统设计（包括物理系统或社会系统），都可

以应用计算机辅助优化技术进行设计优化。

随着CAO问题日益复杂,CAO技术也得到不断发展,其主要发展趋势为:

(1) 在优化技术方面,从单目标发展到多目标优化(multi-objective optimization, MOO),从单学科发展到多学科优化(multi-disciplinary optimization, MDO)。

(2) 在支撑环境方面,CAO建模环境将采用模型驱动体系结构MDA。

(3) 在计算环境方面,CAO从单一计算机发展到多机分布式并行计算。特别是,协同优化(collaborative optimization, CO)技术得到广泛重视。协同优化将一个复杂问题分解成多个较简单的子问题,在分布式仿真、高性能计算技术的支持下,利用异构硬件、软件资源分布式建模与计算,以两层优化结构实现对大系统、复杂问题的计算机辅助优化设计与求解。

参考文献

1. 刘惟信. 机械最优化设计. 2版. 北京:清华大学出版社,1997

2. 阳明盛,罗长童. 最优化原理、方法及求解软件. 北京:科学出版社,2006 (王威　肖田元)

jisuanji fuzhu zhizao

计算机辅助制造(computer aided manufacturing,CAM)　利用计算机和产品制造知识对产品从生产准备到完成的全过程进行作业规划、管理和控制的方法与技术。计算机辅助制造(CAM)技术主要包括生产设备的数字控制与编程、零件加工与产品装配过程的建模与仿真、生产过程的信息采集与处理、产品质量信息的采集与处理、生产计划的生成与管理、设备与物料流的调度与管理、制造系统的运行分析与优化等内容,是计算机技术、检测技术、自动控制、产品制造理论与技术等知识的综合,涉及知识表达与处理、图形图像处理、数据结构与数据库、网络与通信、系统建模与仿真、系统运行控制等知识领域。

计算机辅助制造(CAM)技术与数控技术的发展密切相关。1952年由美国麻省理工学院研制成功的第一台数控(numerical control, NC)机床和1955年提出的自动编程语言(APT)奠定了CAM的雏形。CAM的基本体系构架形成于20世纪50—60年代,其代表有1958年研制的自动换刀镗铣加工中心(machining center—MC)、1962年基于机床数控技术的第一台工业机器人和直接数字控制系统(direct NC, DNC)、1967年建造的第一条由计算机集中控制的自动化制造系统(即柔性制造系统(flexible manufacturing system, FMS))。20世纪70—80年代,几何造型技术、图形显示技术和数控编程后置处理技术、生产系统建模与优化方法、CAD(**计算机辅助设计**)/CAM集成技术的发展和应用,标志着CAM技术发展到了成熟阶段。

计算机辅助制造(CAM)的主要技术内容为:

(1) 生产设备的数字控制与编程　包括数字控制系统和设备控制程序两大部分。数字控制系统由计算机、伺服驱动器、传感器等硬件和数控系统软件组成。其中,数控系统软件完成设备控制程序指令的检验、解释、插补和设备的加减速控制等任务。设备控制程序是用户针对具体任务编制的控制生产设备(如机器人、机床、夹具等)的专用程序,可采用人工编程和计算机辅助编程两种方法完成编程任务。

(2) 零件加工与产品装配过程的建模　是定义产品在计算机内部表示的数字模型、数字信息以及图形信息的工具,是产品信息化的源头。它为产品设计分析、工程图生成、数控加工编制与加工仿真、数字化加工与装配中的碰撞干涉检查、生产过程管理等提供有关产品的信息描述与表达方法。具体涉及几何信息、物理信息、功能信息、工艺信息、运动学信息等的描述、处理、存储和管理。

(3) 生产过程的信息采集与处理系统　由传感器、二次仪表和计算机数据采集接口电路、计算机等硬件和数据采集处理软件组成,主要完成加工设备状态、刀具状态、加工过程、加工系统运行环境以及与生产任务相关的各类信息的采集、处理和存储任务。

(4) 产品质量信息的采集、处理与管理　是以计算机、网络和数据库为手段,应用计算机的信息处理和数据存储、管理能力,协助人们完成质量管理、质量保证和质量控制中的各项工作。具体涉及质量计划的制定、质量检验与数据采集、质量评价与控制、质量综合管理等内容。

此外,更广义的理解,计算机辅助制造还包括计算机辅助生产过程管理、制造执行系统MES、生产设备与物料流的调度与控制、制造系统的运行分析与优化等内容。

计算机辅助制造(CAM)技术的迅速发展及其在制造企业的广泛应用,已经显示出其强大的生命力,并成为增强企业产品竞争能力的强有力手段。采用CAM技术,可以减少原材料消耗,降低成本;提

高产品加工精度,获得稳定的加工质量;操作过程容易实现自动化,生产效率高,减轻劳动强度;生产准备周期短,可大量节省专用工艺装备,适应产品快速更新换代的需要;CAM 与 CAD 衔接紧密,可直接从产品的数字模型产生加工指令,保证零件具有精确的协调和互换性。一般而言,生产对象的形状越复杂,加工精度要求越高,设计更改越频繁,生产批量越小,CAM 的优势就越容易得到发挥。CAM 系统是计算机集成制造系统(CIMS)、现代自动化制造系统的重要组成部分,在机械制造和其他制造行业中具有广阔的应用前景,并给企业带来了巨大的经济效益。

计算机辅助制造(CAM)的发展趋势主要体现在集成、开放、智能、友好、精密、高速等几个方面。集成主要是指 CAM 系统更加紧密和友善地与计算机集成制造系统其他子系统连接起来,实现数据和资源共享;开放主要是指 CAM 系统的体系结构和软件接口对用户更加透明,提供用户扩展系统功能的能力,使得用户的知识和经验能够更好地融入到 CAM 系统中或实现系统的用户定制;智能主要是指将**人工智能**等应用于 CAM 系统中,使其具有人类专家的经验和知识,具有学习、推理、联想和判断功能,从而解决那些目前必须由人类专家才能解决的制造问题;友好主要是指应用**虚拟现实**等技术手段改进和完善 CAM 系统的人机交换方式和环境,进一步方便用户使用 CAM 系统;精密和高速则是指 CAM 系统能够适应和满足设备精度、运行速度等性能不断提高的要求。

参考文献

1. 姚英学,蔡颖. 计算机辅助设计与制造. 北京:高等教育出版社,2009

2. 吴锡英. 计算机辅助机械制造. 南京:东南大学出版社,1990

3. Korea Y. Computer control of manufacturing system. McGraw Hill Book Company, 1983 (姚英学)

jisuanji fuzhu zhiliang kongzhi

计算机辅助质量控制(computer aided quality control, CAQ) 运用计算机实现产品、过程或服务质量的策划、控制、保证与改进的技术。它旨在提高质量控制的工作绩效,保证或改善产品、过程或服务质量,降低缺陷率或不合格率,提高顾客的满意程度,进而提升市场竞争能力。

20 世纪 50 年代,人们就开始使用计算机进行现场质量控制。用计算机对现场质检数据进行统计计算,绘制和分析各种控制图,使原本由人工进行的、繁杂、单调的统计工作变得简单易行,大大提高了工作效率。随着计算机技术的发展与普及推动了质量检测仪器与设备的自动化,相继出现了计算机控制的坐标测量机、机器视觉检测、测量机器人等,促进了计算机辅助检测(CAT)和自动检测技术的发展,为 CAQ 的形成与发展奠定了基础。同时,由于**计算机辅助设计**(CAD)、**计算机辅助工艺规划**(CAPP)、**计算机辅助制造**(CAM)及**计算机集成制造系统**(CIMS)的发展,使计算机在质量管理活动中的应用范围不断扩大,CAQ 不再仅仅局限于现场质量控制,在产品设计、市场研究、供销服务等各环节也都得到了系统的应用。80 年代中、后期,CAQ 逐渐发展成为覆盖产品全生命周期的综合性技术,实现了从关注质量数据,到关注质量信息,再到关注质量知识的挖掘与应用阶段的飞跃。目前,CAQ 已进入了全面应用阶段,相继出现了多种商品化 CAQ 产品,取得良好的经济效益。

CAQ 的基本内容包括:

(1) 质量检测和数据采集 质量检测是质量控制的不可或缺重要环节,计算机辅助检测、在线检测或自动检测技术具有检测效率高、测量精度高、人为误差小、可进行误差补偿、反馈控制和自诊断等特点,可实现产品质量的 100% 检测。质量数据是指导质量策划、控制、保证与改进活动的基础。借助于计算机进行数据采集,具有数据传输快、出错率小、成本低、数据存储量大、便于及时分析与处理等优点,因而得到广泛应用。数据采集方式主要有:由人工检测或观察得到的质量数据、质量信息输入到计算机存储;借助于计算机辅助检测、试验,将需要采集的非电量转化为电量,并通过放大、调理、模数转化,直接传输给计算机储存与分析处理。此外,条码技术、无线射频识别(RFID)技术在质量数据采集中发挥着日益重要的作用。在计算机集成制造系统 CIMS 环境下,CAQ 系统常采用多台计算机组成数据采集网络或直接利用因特网来进行质量数据的采集、储存、传输与共享。

(2) 统计质量分析 使用计算机对质量数据进行统计分析,不仅可以大大提高工作效率,还可以降低对使用者的知识技能要求,获得更为准确的分析结果。常用的统计分析软件有 Mintab、SPSS、SAS 等。此外,频谱分析,时间序列分析、**模式识别**、**专家系统**和**人工神经网络**等信息处理技术也被广泛用于质量数据分析,对产品、过程或服务质量波动状况或

趋势进行分析、评估和预测，从而实施有效的控制、改进与完善。

（3）过程质量控制　贯穿于市场分析、产品设计、制造、检验、试验、销售和服务等整个产品生命周期的过程质量控制，是保持和改善产品质量的最佳途径。使用计算机进行统计过程控制（SPC）、工序诊断、误差补偿、在线监控、预防性控制以及过程质量管理等，可及时地防范和解决各种过程质量问题，特别是制造过程的质量波动问题，使工序处于受控状态，从而保证产品质量的符合性与一致性。网络化的CAQ系统还可将制造过程的质量信息及时地反馈给设计者，以利于产品、工艺及制造系统的整体优化。

（4）质量信息管理　利用计算机及数据库技术实现量大面广的质量信息的采集、传输、存储和使用，支持CAQ与CAD/CAM/CAPP等其他系统的信息集成与共享，使质量系统与技术系统有机地集成在一起。质量信息管理系统可辅助制订质量计划，确定“受控”参数，执行质量数据采集；可将操作指令、计划、工具、检验计量、鉴定要求、质量成本、质量标准等信息储存于数据库中，并定期地更新质量数据，以便于查询与使用；定期分析与处理质量数据，并提供各种质量报表；进行质量信息或质量文件的传输与发布；提供质量设计、评价与决策、设计更改、问题管理和质量预测等辅助支持；提供检验记录、产品缺陷及其责任者的信息追溯等。质量信息管理系统具有综合和调度功能，在CAQ中起着十分重要的作用。

CAQ实现了企业质量控制、质量保证、质量管理的信息化与自动化，适用于大多数类型的产品设计、制造或服务过程，尤其是在计算机集成制造系统CIMS和企业信息化中发挥着越来越大的作用。

计算机辅助质量控制呈现出网络化、智能化发展趋势：

（1）网络化　以internet/intranet/extranet为平台的计算机辅助质量系统，从支持企业内部的质量活动，扩展到整个供应链系统，建立制造商、供应商与分销商等供应链系统之间的质量数据通信、信息集成与知识共享机制，建立网络环境下企业级的质量管理体系和集成化质量管理体系，实现对整个企业乃至供应链的全面质量监控与管理。

（2）智能化　无论是质量策划、质量控制、质量保证，还是质量改进活动，都需要大量的数据、信息与知识支持。借助于人工智能技术帮助CAQ系统实现在恰当的时间、恰当的地点、给恰当的人员提供正确的质量知识支持，一直是CAQ的努力方向。将数据挖掘、**人工神经网络**、**专家系统**等智能化技术应用到质量问题决策、质量信息综合处理、加工质量分析与诊断、检测规程生成以及制造过程监控与诊断等方面的研究已成为当前的热点，并取得较大进展。

参考文献

1. Tannock J D T. Automating qualigy systems. London: Chapman & Hall, 1992

2. 林志航. 计算机辅助质量系统. 北京：机械工业出版社，1997

（余忠华）

jisuanji guzhang geli

计算机故障隔离（fault isolation of computers）　指在计算机软、硬件发生故障（故障诊断）时，通过隔离的手段将引起故障的各种组件、设备或者软件模块隔离开来，其目的是通过隔离手段避免故障再次发生。故障隔离可以作为电路层的硬件设计的一部分，也可以作为整个系统设计的组成部分。计算机故障隔离与计算机故障检测密切相关，故障检测多由于故障已经发生，从而监测到故障本身；而故障隔离则更注重故障发生的原因与确切位置。

根据计算机能够检测到的故障的不同，可以从整体上分为计算机硬件故障和软件故障，所以计算机故障隔离也可根据故障对象的不同分为硬件故障隔离和软件故障隔离。大量的技术被应用于计算机软件的故障隔离。例如，可以通过在不同的地址空间运行程序模块，将程序模块隔离开来；可以通过校验中间过程输出和记录操作步骤日志来手工查找造成程序无法正常运行的进程。在网络环境中，则可以通过在各个节点中放置大量的智能代理工具，通过大量采集实时网络流量数据进而达到监测故障发生的目的。而针对计算机硬件的故障隔离，可以采用硬件禁用、删除或者逐步添加至最小系统等方法有效实现。

计算机故障检测的基本手段通常有两种：基于故障检测模型和基于信号处理过程，因此针对不同的检测手段也对应产生了不同的隔离方法。

计算机故障隔离的实现一般可以通过在系统中嵌入测试回路，或者将操作划分成可进行独立监控的区域或组件来实现，实现了故障隔离的部分可以被手工或自动替换。

参考文献

1. Wang Yanyan, Lu Changhua. Research on fault

diagnosis of mixed-signal circuit. Modern Electronic Technique, 2005

2. Clark D D. Fault isolation and recovery. MIT Press. 1982

（余辰）

jisuanji guzhang xiufu

计算机故障修复（recovery from the failure of computers） 一种解决计算机故障的过程。它包含故障潜在原因评估、排除及确定故障原因、修复计算机等几个步骤。

计算机故障修复需要识别故障的表征。故障是计算机运行过程中发生的一种非预期异常行为。根据故障发生根源，计算机故障包括计算机内部故障（如硬件和软件）、外部的威胁造成的故障（如病毒和黑客）、电源以及自然灾害引发的故障。根据故障发生频率，计算机故障包括可重现故障和间歇性故障。可重现故障是可重复发生的故障，而间歇性故障是仅在特定的不为人所知的条件下发生。由于目前没有有效的工具能够重现间歇性故障，间歇性故障相对来说难修复。

计算机故障修复通常需要系统地寻找导致计算机产生故障的根源。一般来说，根据经验可以评估故障潜在原因。计算机故障的潜在原因可能是计算机系统内在及周边环境中最近的变化，但是，这种内在或外部变化与计算机故障并不必然存在因果关系，可能只是巧合。需要隔离、排除各种潜在原因，确定最有可能原因。

计算机故障原因排除的基本原则是，首先从最简单、最有可能出现的问题开始。故障排除的核心原则之一是重现问题，以便隔离和解决故障。因此，故障修复需要投入大量工作来重现故障，以确定诱发故障的原因。

确定故障的原因可以采用多种技术，如统计方法、压力测试等以重现故障，找出故障与原因的内在关联。高效的故障排除需要清楚地了解计算机系统的预期行为和故障症状。确定故障的潜在原因，并制订若干测试来排除潜在的原因。

故障修复过程可以通过检查列表、流程图等工具来组织。一旦确定故障原因后，那么可以制订修复方案来隔离故障或修理或更换故障部件，使得计算机能够恢复到原始工作状态。

计算机故障修复技术主要包括两大类：一类来自于系统工程、机械、电子等领域，另一类是来自于计算机领域。前者包括故障模式与影响分析（failure mode and effects analysis，FMEA）和故障树分析（fault tree analysis，FTA）等故障分析技术。故障模式与影响分析是一种操作规程，旨在对系统范围内潜在的故障模式加以分析，以便按照严重程度加以分类，或者确定故障对于该系统的影响。而故障树分析通过对可能造成系统故障的各种可能因素进行分析，从而确定系统故障原因的各种可能组合方式和发生概率。由于计算机传递和处理的数据是对人不可见的二进制信息流，为了帮助人修复故障，目前已经研发很多技术和工具。这些工具可收集和分析计算机系统内部状态信息，如日志、运行轨迹、内存镜像、处理器状态、磁盘、网络等，从而提高计算机故障修复效率。此外，计算机取证技术和灾备技术等同样可用于计算机故障修复。

参考文献

1. IEEE Standard Computer Dictionary, 610. 5, 1990

2. 尼尔森，等. 杜江，白志，刘刚，译. 计算机取证调查指南. 重庆：重庆大学出版社，2009

（袁平鹏）

jisuanji guzhang zhenduan

计算机故障诊断（failure diagnosis of computers） 为了确定计算机故障原因并防止其再次发生而分析现象、数据以及定位故障的过程。在故障分析过程中，可采用多样的故障检测方法及技术手段，收集故障部分的数据和信息，定位故障以便用于故障原因的后续分析。

计算机故障现象主要包括：①加装设备或应用后，系统运行不稳定，如死机或重启等；②用户所加装的设备不能正常工作；③用户开发的应用不能正常工作；④用户所需的配置不能满足等。

计算机故障原因大致可分为两类，即硬件故障和软件故障。硬件故障一般由**中央处理器**（CPU）、主板、显卡、声卡、网卡、硬盘、显示器等硬件的某些使用参数超过了元件所允许的极限值而产生。计算机网络技术的进步，在带给人们巨大经济效益的同时，也为计算机病毒的传播提供了更加便捷的途径，使得软件系统易遭受计算机病毒攻击，从而造成计算机系统故障。另外，使用者的不当操作也会使得软件系统无法正常工作，造成计算机系统出现故障。软件故障主要分为以下几类，即基本输入输出系统（BIOS）故障、病毒攻击、操作系统引导故障、用户对计算机的误操作等。

计算机故障诊断方法一般遵循由外到内、先软（件）后硬（件）、由外设至主机、由简入繁、主次分明等原则，故障诊断的处理方法及技术手段主要有：故障模拟、可测性设计（DFT）、加电自检法、直接观察法、最小系统法、隔离法、替换法、升降温法、操作系统诊断以及病毒查杀等。

故障模拟　为确定所选取的测试码（向被测对象输入的激励信号）的故障覆盖率（被检测到的故障数占系统中所有可能发生故障总数的百分比）和提高测试效率而采用的技术手段。包括：故障检测、生成字典进行故障定位以及对被测对象的故障模拟分析等。目前已广泛应用于数字系统，尤其是大规模集成电路的研制中。

可测性设计　在电路设计阶段就考虑采用一些便于测试、可降低测试难度和测试工作量的技术及方法，从而设计以较小的综合代价完成预定的故障测试任务的系统。

加电自检法　接通电源，计算机首先自动运行主板 BIOS 芯片固化的程序，通常称为上电自检（power on self test，POST）。完整的加电自检法包括对 CPU、主板、内存、只读存储器（ROM）BIOS 的测试；互补金属氧化物半导体（CMOS）中系统配置的校验；初始化视频控制器，测试视频内存、检验视频信号和同步信号；对键盘、软驱、硬盘及只读碟（CD-ROM）子系统做检查；对并行口和串行口进行检查。

直接观察法　通过看、听、摸、闻等方式检查比较典型或比较明显的故障。

最小系统法　拔去怀疑有故障的板卡和设备，并根据机器在此前和此后的运行情况对比，判断并定位故障所在。拔插板卡和设备的基本要求是保留系统工作的最小配置，以便缩小故障的范围。

隔离法　将可能妨碍故障判断的硬件或软件屏蔽起来的一种判断方法。它也可用来将怀疑相互冲突的硬件、软件隔离开以判断故障是否发生变化的一种方法。

替换法　用好的部件去代替可能有故障的部件，以判断故障现象是否消失的一种维修方法。好的部件可以是同型号的，也可以是不同型号的。

比较法　比较法与替换法类似，即用好的部件与怀疑有故障的部件进行外观、配置、运行现象等方面的比较，也可在两台计算机间进行比较，以判断故障计算机在环境设置、硬件配置方面的不同，从而找出故障部位。

升降温法　升温法是设法降低计算机的通风性能，靠计算机本身的发热来升高温度，检查计算机能否温度升高而产生毛病；降温法则是选择环境温度较低的时间段，停机 12～24 小时或以上或用外部冷却设备加快降低温度，检查能否因温度降低使得故障消失或发生改变。

操作系统诊断及修复　针对具体的操作系统（如 Windows）的故障，如磁盘空间骤减、网络连接故障、注册表信息篡改、系统引导失败、应用软件与操作系统不兼容以及蓝屏等采取的技术手段及方法。

病毒查杀　运行杀毒软件，修复因感染病毒而招致破坏的文件从而恢复软件系统。

参考文献

1. 伍旦初. 计算机组装和维护教程. 北京：电子工业出版社，2008

2. Abramovici M，Breuer M，Friedman A. Digital systems testing and testable design. Computer Science Press，1991

（韩建军　徐拾义）

jisuanji guocheng kongzhi

计算机过程控制（computerized process control）　使用计算机系统实现生产过程的在线监视、操作指导、控制和管理的技术。这里的“生产过程”通常是指把原材料转变成产品，并具有一定生产规模的过程。这种生产过程生产方式可分两大类，一是高度连续（或批处理）的生产过程，例如炼油、石油化工、冶金、发电、化工、轻工、水处理等；另一类是离散制造过程，例如机械加工、汽车制造等。生产过程的特点是实时性，生产过程中必须做到安全生产，稳定运行，单位产品的消耗和废物要少。因此，生产过程自动化是现代工业必不可少的内容。“在线”指的是计算机系统与过程装置之间有直接联系，能实时地从过程输入测量信息和向它输出控制信息，使过程按操作人员的预定要求或随机要求实现变化。

现今，计算机是生产过程自动化不可缺少的工具。不过，用于生产过程控制的计算机必须具有高可靠性，其平均故障间隔时间（MTBF）一般要达到几万小时以上。与普通计算机不一样，作为控制计算机，它必须具备与生产过程各种传感器、执行器相互连接的接口设备，例如，模拟量或数字量输入/输出接口卡件等。在软件方面，要提供为过程控制所用的专用组态软件以及便于用户自己编程的简单、易学专用语言。目前主要的过程控制计算机有**集散控制系统**（DCS）、可编程逻辑控制器（PLC）、工业控

制微机系统和**现场总线控制系统**(FCS)。在连续生产过程中多采用集散控制系统(DCS);而在离散制造过程中多应用可编程逻辑控制器(PLC)。

计算机在生产过程中的应用已经走过50多年历史了。至今,在生产过程控制和管理中都广泛应用计算机,形成了工业生产过程控制和管理的综合自动化递阶结构,如图1所示。由图可见,对生产过程而言,计算机主要实现常规控制、先进控制、监督控制和决策调度四个方面的功能。

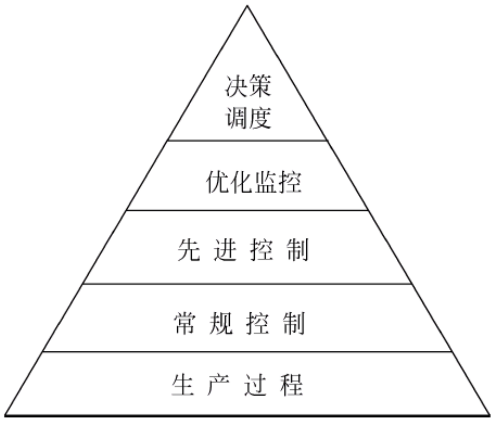


图1　企业生产过程管理与控制的递阶结构

常规控制功能　即基础自动化,一般由DCS来实现。计算机主要完成的功能包括:①实时生产过程信息的采集、显示、记录、报警;②生产过程安全控制,其中有信号连锁控制、生产过程开停车控制;③生产过程工艺参数(如温度、压力、流量、液面等)的稳定控制,其中有简单的PID控制、串级控制、比值控制和前馈控制;④人机交互操作界面,手动与自动切换等。

先进控制功能　常规控制是以单个工艺参数组成的单回路控制。而先进控制是面向整个单元操作或流程的多变量稳定控制。目前常用的是多变量模型预测控制软件包。通常由工业控制微机和实时数据库与DCS相结合来实现。

优化监控功能　这一层的目标是为先进控制层(或常规控制层)提供优化的控制设定值。它的功能是根据调度指令、实时生产过程信息和其他约束条件,如原材料品质和数量、产品规格和生产设备等,采用最优化方法,求解生产过程优化方案和提供给先进控制层的优化目标设定值。

决策调度功能　一个企业年度、季度和月度计划目标的实施是通过决策调度层进行优化计算给出一段时间内,如三天、一周或半个月的生产作业计划,其中包括产品品种、规格、数量和原材料使用计划以及公用工程需求计划。这一层的实施是离线进行的,通常属于管理信息系统范畴,其基础条件是实时数据库和关系数据中有关生产过程、市场等信息。这层的主要功能是把离线数据与实时数据相结合变成在线可应用的数据,即优化结果的生产作业计划数据为优化监控层提供调度指令。

从20世纪90年代起,随着计算机和网络系统在生产过程中的广泛应用,工业企业生产过程,开始逐步实现过程自动化(PA)、工厂自动化(FA)、计算机集成过程控制(CIPS)和计算机集成制造系统(CIMS)。

参考文献

1. 王锦标,方崇智. 过程计算机控制. 北京:清华大学出版社,1992

2. Love J. Process automation handbook: a guide to theory and practice. Springer, 2007

3. 王树青,等. 工业过程控制工程. 北京:化学工业出版社,2002

(赵豫红　王树青)

jisuanji guocheng kongzhi fangshi

计算机过程控制方式(computerized process control manner)　计算机在生产过程控制中应用的各种结构方式。根据计算机在生产过程中应用的历史过程和通常的术语,计算机控制结构方式可分为操作指导、直接数字控制、集散控制、可编程逻辑控制、监督控制、管理信息系统和计算机集成制造七种形式。

操作指导　早期的计算机因可靠性问题,只能用于离线操作指导的方式。计算机通过实时采集生产过程数据,经过处理提供操作指导和监视生产过程。

直接数字控制(DDC)　到了20世纪60年代,计算机的可靠性有所提高,人们将计算机直接用于控制生产过程,即直接数字控制(DDC),从而实现一台计算机对整个生产过程信息的显示、记录、报警和控制的全部功能。

因为计算机的可靠性问题,当初只在不危险的工厂中试验。现在,大量的数字控制装置已成为控制回路的一部分,实现各种直接数字控制的目的。

集散控制(DCS)　早在1975年,Honewell公司首先推出基于微处理器的集散控制系统(DCS)——TDC 2000,从而开创了计算机真正在工业生产过程中应用的新天地,这种集散控制系统的结构包括过

程接口单元(PIU)、过程控制单元(PCU)和操作员控制站(OCS)。这些单元用高速公路(总线)相互连接,形成了多节点、分散式计算机控制系统。

高速公路(总线),为过程控制单元和操作员控制站提供实时快速的通信,将PCU信息提供给OCS,而操作员的指令通过OCS可传递给PCU。为了确保可靠的传递这种十分重要的信息,这种高速公路(总线)通常都采用双冗余的通信线路,每个PCU和OCS单元同时与两个通信线相连,即使工作的通信线出了故障,另一路将自动切换上,保证通信线路正常工作,从而提高了计算控制系统的可靠性。

另外还有其他模块单元,如历史模块(HM)和应用模块(AM)也连接在高速公路(总线)上。历史模块HM可存储大量来自PCU的生产过程历史数据,并为过程控制单元PCU提供历史数据信息。同样应用模块AM,可提供比PCU功能更强的高级控制软件包及信息处理功能,例如,可为先进控制、优化控制、随机统计过程控制、专家系统等提供支持。(参见**集散控制系统**)。

可编程逻辑控制(PLC) 历史上离散制造生产过程大量的离散输入输出信号和逻辑关系运算是采用硬件形式的机电逻辑电路来实现的。随着制造业规模的不断扩大,机电逻辑电路不能适应制造业的需求,因此基于微处理器的可编程逻辑控制器已成为现代制造业的自动化工具。可编程逻辑控制系统的结构类似于集散控制系统(DCS),由过程接口单元(PIU)处理工厂的输入输出信号,由PLC来处理逻辑运算控制,PLC通过高速公路(总线)和操作员控制站OCS相互连接。PLC也可作为批处理生产过程自动化装置以及DCS和优化监控系统的前端设备(参见**可编程控制器**)。

监督控制和数据采集(SCAD) 监督控制和数据采集实际上是计算应用的一个术语而不是计算机控制结构。早期的SCAD系统是一种离线的计算机应用,而今则大不一样。一般来说,SCAD系统是用于监视和数据采集,它的控制功能是优化设定单回路控制系统或先进控制系统的设定值,因此,这种系统具有大量的信号输入和少量的信号输出。

从硬件系统来看,SCAD系统只是在DCS、PLC、单回路控制器(SLC)等基础控制系统上,增加了一层**局域网**(LAN)和个人计算机系统。基础控制系统通过**网关**(GW)与局域网相连。

管理信息系统(MIS) 一个典型的管理信息系统包含一个或多个“主计算机”作为文件处理服务,通过网关(GW)与DCS的高速公路(总线)及现场输入输出接口(I/O)相连。所有的DCS和其他系统的信息都汇集到主计算机进行存储和处理。

所有信息存在数据库里,因此在对数据进行操作和处理时,不与DCS直接相连。一个管理信息系统(MIS)能够在线对全厂范围内基于模型的各种需求进行计算,例如计算生产过程效率、工厂仓储库存和公用工程消耗等。(参见**管理信息系统**)。

计算机集成制造(CIM) 计算机集成制造是**管理信息系统**的进一步拓展和集成。CIM与MIS最大的差别在于CIM是用于全厂控制的目标,从主计算机到现场控制单元的信息流是双向的,例如为了达到生产能力最大或使性能最优而生产成本最少,及时进行生产调度。CIM的优化计算是实时的,因此,必须考虑生产过程能力的需求、原材料的可用性以及工厂公用工程的能力等。CIM通常采用集成的方法,在集成体系中,PCU和OCS通过高速公路互相传送数据,而各个子集的功能是彼此独立的。因此,网络的作用只是传递各自需要的数据。(参见**计算机集成制造系统**)。

参考文献

1. Love J. Process automation handbook: a guide to theory and practice. Springer, 2007

2. 谢剑英,等. 微型计算机控制技术. 3版. 北京:国防工业出版社, 2001

3. 王锦标. 计算机控制系统. 2版. 北京:清华大学出版社,2008

(赵豫红 王树青)

jisuanji jifang sheshi

计算机机房设施(computer room facility)

保证计算机在机房安全可靠运行所配置的各种设施。计算机机房指安装各类计算机系统及相关设备的专用场所。

机房环境条件

温度与湿度 它们是计算机机房最重要的环境条件,对计算机可靠运行和寿命均有显著影响。温度偏高导致元器件散热困难,容易造成元器件损坏及工作不稳定,而且加速元器件的腐蚀和磨损而影响其寿命。温度过低则导致部分敏感的机械部件由于膨胀系数不同而运转失误。温度梯度也需控制在一定范围,因温度变化速度过快会造成某些敏感部件的运转失误。温度条件还应考虑机房工作人员舒适性需要。为此,大多数机房规定的温度范围为18~25 ℃,冬天可靠近下限,夏天则靠近上限。温

度梯度的范围大致为 5 ~ 10 ℃/h。

湿度过高容易造成高密度组装的元器件表面或接插件的导电体之间短路，还会使某些易吸湿物质(如纸张)运动不顺畅。湿度过低会使机房中容易产生静电高压，不仅影响计算机正常运行，还易损坏集成电路。机房适宜的相对湿度为 50% ±10%。

洁净度　指空气中单位体积内的尘埃含量。它也是计算机机房的重要指标。灰尘对磁盘机、**磁带机**及**软磁盘驱动器**等精密机械会造成严重危害。例如**硬磁盘驱动器**磁头和碟片之间的缝隙只有零点几微米，如缝隙中沾染了灰尘就可能发生存取数据错误并会导致磁头磨损或划伤碟片。灰尘如积聚在高密度电路表面或接插部位，则既容易造成吸湿短路也可能引起接触不良。空气过滤器中积聚过多灰尘则导致通风不畅使机器过热，机械转动部分落入灰尘会使设备磨损加剧。机房洁净度以单位容积中尘埃粒数衡量，A 级机房要求此数小于或等于 3500 粒/ft^3，B 级机房要求此数小于或等于 10 000 粒/ft^3(尘埃粒径大于或等于 0.5 μm，1 ft^3 = 0.0283 m^3)。

静电与电磁干扰　静电是由于摩擦生电、感应或辐射造成的电荷积累产生的一种高压电场。而计算机设备所包含的半导体器件如金属-氧化物-半导体(MOS)电路及其他高输入阻抗电路对静电非常敏感，这些电路易受静电干扰导致工作失效，甚至被静电高压损坏。因此机房应采取防静电措施。

各种来自外界(如雷电、邻近的高强度无线电波及强高压电器设备)的电磁干扰会影响计算机中一些高灵敏度的弱信号放大器正常工作。机房应采取措施予以防范，如确保设备接地良好，必要时对某些关键装置进行屏蔽防护。按照 GB 2887—1982 规定，机房内无线电干扰场强，在频率范围为 0.15 ~ 500 MHz 时应不大于 126 dB，磁场干扰场强不大于 800 A/m。

电源与接地　机房必须有良好的供电系统以保证计算机正常运行。对于必须连续不间断运行的计算机系统应采用双路供电的方式，避免任何一路因偶然故障导致系统瘫痪。机房的供电交流电源应有较高的电压和频率稳定度。按照 GB 2887—1982 规定，A 级电压相对误差范围是 −5% ~ +5%，B 级电压相对误差范围是 −10% ~ +7%，C 级电压相对误差范围是 −10% ~ +10%；A 级频率误差范围是 −0.05 ~ +0.05 Hz，B 级频率误差范围是 −0.5 ~ +0.5 Hz，C 级频率误差范围是 −1 ~ +1 Hz。

计算机机房有多种接地线，即交流电源地线、直流电源地线及安全接地线等。这些地线有的可以共用，有的则必须分开。共用地线简便但可能造成交流信号对直流地线的干扰而影响计算机正常工作。目前较常用的方法是直流地与安全地共用而与交流地线分开以避免干扰。接地电阻数值应尽量小，一般直流地与安全地的接地电阻小于 1 ~ 2 Ω，交流地接地电阻小于 2 ~ 4 Ω。

计算机机房设施

空调通风设施　为保持机房的洁净度并将温度及湿度控制在特定范围内，机房采用封闭的环境并借助空调设施进行通风。目前大多数计算机机房都安装了专用空调装置，以便根据季节变化及计算机的散热量灵活调节空调机容量，使机房温湿度控制在所需范围内。

冷却设施　随着计算机系统处理能力的迅速提高，其能耗越来越大，特别是高性能计算机使用的服务器电路芯片，能耗很高产生的热量大加以组装非常紧凑，使得局部空间功率密度很高，有的计算机机柜的功耗可达几十个千瓦，一般空调设施已难以达到散热的需要，而必须采用液体冷却装置，把专用的冷却液体直接送到机柜内部，通过热交换装置达到冷却降温的目的。

不间断电源(UPS)　它具有稳频、稳压功能。供电线路通过它将交流电送到计算机，不仅提高了电源稳定性而且当外界供电突然中断时，可借助后援电池保持系统继续供电一段时间(根据系统的需要设置，通常为十几分钟到几个小时)，使维护人员可及时采取措施保存系统运行的现场状态信息和其他重要数据，保证系统信息的安全性与完整性。

防静电地板　机房地板宜采用在建筑物原有地板上架高 300 cm 左右的防静电地板，这样可防止在机房中产生严重的静电高压，而且地板下的空间可用作通风回路和各种电源、信号线电缆以及地线的铺设路径。

防火设施　计算机机房四周墙壁及门窗、天花板均应采用防火材料。必要时装置高温和烟雾敏感探测装置和自动报警设备，对机房各个角落进行探测，以便一旦出现可能引起火灾的因素能及时发觉并采取措施予以消除。条件许可时应装置自动灭火设施。

照明　提供工作人员维护计算机的基本条件，照度应按照 GB 2887—1982 标准执行。除正常照明设施外，机房应设置事故照明装置，一旦供电中断时保证机房照明。

其他设施 其他必要的辅助设施包括：保证机房洁净度及清新空气的装置（如进入机房新空气的过滤装置、负离子发生器等），监测机房环境条件的装置（如连续型温湿度记录仪），防灾装置（如防地震的机柜锁定装置、防电击的避雷设施、防止过湿和漏水的水敏传感器以及紧急灭火设施等）。

今后的计算机机房设施将朝智能化方向发展，各种环境条件的提供与保证均采用自动监测、自动控制。考虑到计算机系统本身的脆弱性及计算机犯罪的严重性，计算机机房的相应安全保险设施将日趋完善。

参考文献

劳诚信，姜国雄，等. 计算机安全指南. 北京：清华大学出版社，1993 （苏振泽）

jisuanji jicheng zhizao xitong

计算机集成制造系统（computer integrated manufacturing systems，CIMS） 信息时代一种组织、管理企业生产的哲理和技术。CIMS 综合运用计算机信息处理技术和生产技术，通过对制造企业业务过程（包括市场分析、产品设计、计划管理、加工制造、销售服务等）的信息集成、过程优化及资源优化，实现组织、经营和技术等要素的集成，提高企业的市场应变能力和竞争能力。CIMS 是信息技术、现代管理技术和制造技术相结合的综合性技术，实现制造生产过程中物流、信息流、价值流的集成和优化运行，通过加强企业产品开发的 T（时间）、Q（质量）、C（成本）、S（服务）、E（环境），使企业获得最大总体效益。

发展沿革

计算机集成制造（computer integrated manufacturing，CIM）的概念是美国 J. Harrington 于 1973 年提出的。CIM 技术的形成是由于企业竞争环境日益加剧和计算机技术、信息处理技术、自动化技术的不断发展与相互作用的结果。CIM 技术主要来源于三个方面：计算机辅助设计、计算机辅助管理及制造自动化。在**计算机辅助设计**（CAD）方面，美国于 1963 年研制成功第一个**几何造型** CAD 系统。20 世纪 60 年代末，**计算机辅助工艺规划**（CAPP）、**计算机辅助制造**（CAM）、计算机辅助工程（CAE）等陆续得到发展。在计算机辅助管理方面，美国 IBM 公司于 1968 年提出了第一个生产信息与管理系统（PICS）。其后，出现了功能日益完善的**管理信息系统**（MIS），如物料需求计划（MRP）、制造资源计划（MRP Ⅱ）等。在制造自动化方面，50 年代末，出现了第一台数控机床。60—70 年代，各种数控机床、柔性制造系统（FMS）等不断发展。70 年代，制造业的市场竞争环境日益激烈；为了缩短产品生产周期，提高产品质量，降低产品成本，人们开始重视上述技术集成，陆续实现了 CAD 与 CAM、CAD 与 MIS、CAPP 与 MIS 等局部系统的集成。80 年代，随着计算机、网络、数据库等信息技术的迅速发展，CIM 进入了技术成熟期，在技术和内涵上都得到了很大发展。90 年代，**产品数据管理**（PDM）、**产品生命周期管理**（PLM）、**并行工程**（CE）、精良生产、敏捷制造、**企业资源计划**（ERP）、经营过程重组等新概念、新技术不断涌现，丰富了 CIM 技术的内涵和外延，企业从 CIM 的应用中也获得了巨大效益。

我国从 1986 年起把 CIMS 作为国家高技术研究发展计划（亦称为“863”计划）中的一个主题开始实施。1987 年，我国第一个 CIMS 技术研究实验基地——CIMS 实验工程在清华大学开始建设，并于 1992 年完成，1994 年，获得美国制造工程师协会（SME）颁发的大学领先奖。1989 年开始，陆续有不同行业的数百家企业实施 CIMS，大多获得了明显的效益。1995 年，北京第一机床厂 CIMS 应用示范工程获得美国制造工程师协会颁发的工业领先奖。21 世纪以来，CIMS 技术在我国制造业信息化工程中继续发挥着重要作用。

随着信息技术的进步，CIMS 的内涵也在不断地丰富并向纵深发展。CIMS 的发展大体上经历了三个阶段：信息集成阶段（1980—1990）、过程优化集成阶段（1990—2000）、协同制造阶段（2000 年至今）。

技术内容与系统构成

CIMS 技术内容比较广泛，主要包括 CIMS 总体技术、CIMS 设计自动化技术、CIMS 制造自动化技术、CIMS 管理技术、CIMS 质量保证技术、CIMS 支撑技术等。此外，CIMS 相关技术还包括并行工程、敏捷制造、虚拟制造、企业互操作、网络化制造等。

（1）CIMS 总体技术 从整体上分析复杂的 CIMS，并提供一套指导企业正确实施 CIMS 的方法和工具，具体内容包括：企业运行模式、CIMS 体系结构、企业建模分析和业务设计优化方法、CIMS 实施方法论、CIMS 集成技术和工具、CIMS 标准化与规范等。

（2）CIMS 设计自动化技术 利用计算机软硬

件及网络环境实现产品设计与开发的全过程。具体内容包括：**计算机辅助设计** CAD、**计算机辅助工程** CAE、**计算机辅助工艺规划** CAPP、**产品数据管理**（PDM）、**产品生命周期管理** PLM、集成协同设计、产品优化设计、**产品数据交换标准**等。

（3）CIMS 制造自动化技术　指 CIMS 制造自动化过程和环境中的集成技术和系统技术。具体内容包括：制造自动化系统设计、制造过程的组织和管理（如制造执行系统（MES）、生产计划、调度与控制、自动化物流管理等）、制造装备和制造过程集成控制与监视（如数控和分布式数控、物料装备控制、柔性制造、制造过程检测与监视等）、制造过程建模仿真和虚拟制造、智能制造和绿色制造等。

（4）CIMS 管理技术　通过现代管理模式与计算机管理信息系统支持企业合理系统地管理经营与生产，最大限度地发挥现有设备、资源、人、技术的作用，产生最大企业经济效益。具体内容包括：现代制造企业的管理模式与组织理论、CIM 管理支持系统（如 CIMS 管理信息系统 MIS、决策支持系统 DSS、领导信息系统 EIS 等）、CIM 管理系统方法论等。**企业资源计划** ERP 是典型的 CIMS 管理技术。

（5）CIMS 质量保证技术　对产品全生命周期中质量相关数据进行有效采集、处理、评价和决策，对质量管理、质量保证、质量控制活动进行有效的管理，实现企业质量管理的整体优化。具体内容包括：**计算机辅助质量控制**（CAQ）、质量功能配置（QFD）、集成质量系统（IQS）、面向质量的设计（DFQ）等。

（6）CIMS 支撑技术　为 CIMS 的系统集成、信息集成等提供网络化、集成化计算机信息基础支撑。具体内容包括：**计算机网络**、企业网络设计与实现、**数据库**、**数据仓库**、企业集成框架、企业集成平台技术、企业软总线与企业服务总线、异构分布式系统集成技术等。

根据 CIMS 的典型功能，可以把 CIMS 分为四个功能分系统（即经营管理信息分系统、产品设计自动化分系统、制造自动化分系统和质量保证分系统）和两个支撑分系统（即计算机网络分系统与数据库分系统）。四个功能分系统通过两个支撑分系统有机地实现集成。

（1）经营管理信息分系统　它以制造资源规划 MRPⅡ或企业资源规划 ERP 为核心，具有包括市场预测、经营决策、各级生产计划、生产技术准备、销售、供应、财务、成本、设备、工具以及人力等信息管理功能。

（2）产品设计自动化分系统　它是用计算机来辅助产品设计、工艺规划及制造，即常说的 CAD/CAPP/CAM 系统以及产品数据管理系统 PDM/产品生命周期管理 PLM。

（3）制造自动化分系统　不同制造企业具有不同的制造设备和制造环境。如离散型制造企业的制造自动化系统可能由数控机床、加工中心、清洗机、测量机、各类运输工具、各类仓库等组成。在 CIMS 中，这些设备通过计算机网络及相应支持软件集成在一起，形成制造执行系统 MES，可根据产品的工程技术信息及生产计划，灵活而高效地完成作业调度和制造任务。

（4）质量保证分系统　在计算机系统支持下完成质量决策、质量检测与数据采集、质量评估、控制与跟踪等功能。

（5）数据库支撑分系统　它是覆盖企业全部信息以支持 CIMS 各功能分系统的数据库管理系统。企业数据库系统多采用分布式数据库管理系统。

（6）计算机网络支撑分系统　它是支持 CIMS 的计算机网络系统，一般采用国际标准或工业标准规定的网络协议，支持系统开放、资源共享、异构互联。近年来，在企业计算机网络系统之上，还采用企业集成平台、企业服务总线等支持异构系统的集成。

发展趋势

21 世纪以来，以协同制造为主要特征的企业互操作、企业协同、网络化企业等技术进一步延续了 CIM 技术的发展。协同制造是制造企业基于网络开展协同产品设计、制造、销售、采购、服务等一系列活动的总称，其核心是利用计算机网络，特别是 Internet，实现跨越不同企业环境的信息共享、资源共享、业务过程协同、应用互操作，为企业间异地协同设计、生产、营销、供应链管理等提供技术支撑环境，实现产品商务协同、产品设计协同、产品生产协同、供应链协同、产品服务协同，提高整个产业链和制造群体的竞争力。网络化企业、企业协同、企业互操作、协同产品商务（collaborative product commerce, CPC）、应用服务提供商模式（application server provider, ASP）等都是协同制造的各种形态。协同制造是企业适应信息时代制造全球化发展的信息化新阶段，其进一步发展趋势是制造网格、云制造、企业制造服务化等。

随着 CIMS 在制造企业应用的发展，CIMS 的内涵已经突破 harrington 以及后来的一些定义。本质

上,“CIMS”中的“C”(computer)代表信息化,“M”(manufacturing)代表工业化,“I”(integrated)就是要将企业信息化与工业化相融合,成为一个有机整体的系统“S”(system)。可以说,“两化融合”是我国由世界制造大国发展成为制造强国的重要途径,CIMS技术也将继续发挥重要作用。

参考文献

1. 863/CIMS信息网. 计算机集成制造系统CIMS问答. 北京:兵器工业出版社,1993

2. 吴澄. 现代集成制造系统导论. 北京:清华大学出版社,施普林格出版社,2002

3. 徐晓飞,田雨华,薛劲松. 计算机集成制造系统CIMS知识新解. 北京:兵器工业出版社,2000

(肖田元　徐晓飞)

jisuanji kexue lilun

计算机科学理论(theory of computer science) 计算机科学的基础理论的总称。它以计算机为研究对象,是系统化的基本概念、基本方法、基本定理、基本模型和基本原理体系,主要包括数值计算、离散数学、计算理论和程序理论四部分。

数值计算讨论用于模拟物理过程或社会过程的各种数值算法的设计、分析和使用。早在18世纪与19世纪,高斯、牛顿、傅里叶等著名数学家就研究过数值计算方法,而计算机的诞生更大大促进了数值计算的发展。数值计算涉及的内容颇多,如方程求根、数值逼近、数值微分、数值积分、数值代数、线性代数方程组的数值解法、矩阵特征值计算、微分方程数值解法等。例如,高次代数方程求根的常用方法有二分法、牛顿法、割线法等。数值微分讨论求导数近似值的理论与方法,常用的有有限差分法。数值积分讨论求定积分近似值的理论与方法。梯形法和辛普森法均为世人所熟知。线性代数方程组的数值解法用以求线性代数方程组的数值解,通常有直接法和迭代法两类。高斯消去法为直接法,简单迭代法和赛德尔迭代法均为迭代法。

离散数学泛指数学中讨论离散对象的分支。和连续数学不同,离散数学通常涉及整数系,由于数字计算机是离散机,离散数学的重要性不言而喻。通常认为离散数学包含集合论、图论、组合学、数理逻辑、抽象代数、线性代数、差分方程、离散概率论等学科。图论是研究图的性质的学科。图论中的图并非初等数学中的图,后者只是连续函数的图形,图论中的图却是一组顶点(结点)和一组连接两顶点的边(支)所构成的集合。组合学讨论计算某类对象个数的方法,它在统计学、理论物理、化学、社会科学、通信理论以及计算机科学技术中均有重要作用。多数组合学问题可归结为存在性问题、枚举性问题或选择性问题。数理逻辑研究形式体系。作为其组成部分的命题演算与谓词演算等在计算机科学技术中作用巨大,影响深远。诸如计算机设计、软件开发、程序正确性验证,以及人工智能等领域无不用到数理逻辑。抽象代数讨论离散对象结构,它在计算机科学技术中应用广泛。例如,半群已用于形式语言理论和自动机理论,群在编码理论中有其重要作用。线性代数虽然涉及实变量,但其结构与处理均为离散,因而,也可归为离散数学。此外,差分方程、离散概率论等亦属离散数学。

计算理论主要包括算法、算法学、计算复杂性理论、可计算性理论、自动机理论、形式语言理论等。算法是解题过程的精确描述,它包括有限多条规则,并具有如下性质:第一,将算法作用于特定的输入集,可导致由有限个动作构成的动作序列;第二,该动作序列具有唯一一个初始动作;第三,序列中的每个动作具有一个或多个后继动作(序列中的末尾动作的后继动作可视为空动作);第四,序列或者终止于输出问题的解,或者终止于输出某一陈述,以表明问题对该输入集而言不可解。算法学是系统研究算法的学科。通常包括设计、验证以及分析三部分。设计是创建算法的过程,并研究良好的创建方法;验证在于证明算法的正确性,基本途径是数学归纳法;分析着重确定算法的效用,当一问题有多种算法可用时,则比较其相对效用。计算复杂性确定问题的固有难度,通过研究计算复杂性,可以断定哪些问题是固有困难的。对一些固有困难的优化问题,若需要可退而寻求性能优越的近似算法。算法复杂性是针对特定算法而言的,最佳算法复杂性等于计算复杂性。计算复杂性理论则是用数学方法研究各类问题的计算复杂性的学科。它在计算机科学技术中既有理论意义,又有实用价值。可计算性理论是研究计算的一般性质的数学理论。它通过建立计算的数学模型,精确区分哪些问题是可计算的,哪些问题是不可计算的。计算的过程就是执行算法的过程。可计算性理论主要包括图灵机、丘奇图灵论题、λ演算、原始递归函数、部分递归函数、递归集、递归可枚举集、可判定性等。自动机理论是研究称作自动机的抽象理想机的数学学科。自动机是信息处理设备(如计算机)的抽象。多数自动机都是图灵机的特

例。自动机理论一般包括有限自动机理论、无限自动机理论、概率自动机理论、细胞自动机理论等。形式语言理论是用数学方法研究自然语言(如英语)和人工语言(如程序设计语言)的语法的理论。形式语言就是模拟这些语言的数学工具。它只研究语言的组成规则,不研究语言的含义。内容包括描述工具、文法分类(如乔姆斯基层次)、语言分类,以及各类语言的性质及其间的关系等。

程序理论研究程序的语义性质、语用性质和程序的开发,主要包括程序语义理论、程序语用理论、数据类型理论、程序逻辑理论、程序验证理论、并发程序设计理论和混合程序设计理论等。程序理论和计算理论是计算机科学理论的两大支柱。形式语义理论是用数学方法研究程序语言语义的理论,包括操作语义、公理语义、指称语义以及代数语义等。此外,还有旨在用计算机研究代数演算的“计算机代数”以及用计算机研究数学证明的“计算机数学”等。

参考文献

1. 冯康,等. 数值计算方法. 北京:国防工业出版社,1978

2. Matt J L, Kandel A, Baker T P. Discrete mathematics for computer scientists. Reston, Virginia: Reston Publishing Company, Inc., Prentice Hall Company, 1983

3. Cormen T H, Leiserson C E, Rivest R L, et al. Introduction to algorithms. 3rd ed. Cambridge, MA: The MIT Press, 2009

4. Manna Z. Mathematical theory of computation. New York: McGraw-Hill, 1974

(徐家福 殷建平 祝恩)

jisuanji kongzhi

计算机控制(computer control) 以计算机为核心控制部件并借助一些辅助装置与被控对象相联系,以达到特定控制目的的过程和技术。这里的计算机可以有各种规模,包括从微型到大型的通用或专用计算机。辅助控制装置主要指计算机输入输出接口、检测和执行装置等,它们可以是模拟设备、数字设备或是数模混合设备。计算机与被控对象和辅助控制装置之间的联系,可以是有线方式,如通过电缆的模拟信号或数字信号进行联系,也可以是无线方式,如用红外线、微波、无线电波、光波等进行联系。被控对象的范围很广,包括各种生产过程、机械装置、交通工具、机器人、仪器仪表、家用电器等。控制目的可以是使被控对象的状态或运动过程达到某种要求,也可以是为了达到某种最优化目标。

与一般的自动控制相同,计算机控制可以是闭环的,典型的就是直接数字控制方式。这时计算机不断采集被控对象的状态信息,按照一定的控制策略和步骤处理后,输出控制信息直接影响被控对象。它也可以是开环的,这里有两种方式:一种是计算机只按时间顺序或某种给定的规则影响被控对象,又称为顺序控制系统,主要用在如机械加工、产品安装包装等固定操作程序的行业;另一种是计算机将来自被控对象的信息处理后,只向操作人员提供操作指导信息,称为操作指导方式。

计算机用作控制器的思想萌生于20世纪50年代初。最早,试图在飞行器控制中应用计算机,但是由于当时的通用计算机体积大、能耗高、可靠性差,因此使用了专用机——数字微分分析器。

此后,计算机控制的主要进展是在工业过程控制领域(参见**计算机过程控制**)。美国首先用计算机实现工业生产过程的巡回检测和数据采集。20世纪50年代后期又成功地实现了计算机在线闭环控制。1962年英国实现了计算机直接数字控制,充分利用计算机的高速分时运算能力代替多台常规模拟控制仪表。

20世纪70年代**微型计算机**的诞生,使计算机控制的应用进入一个新阶段。在工业过程控制领域,不久就出现了控制分散、操作管理集中的分散控制系统,也称**集散控制系统** DCS。如美国Honeywell公司于1975年推出的TDC-2000系统。集散控制系统在结构上是由多台计算机及其他设备用通信网络联系在一起的分级分布式计算机控制系统,它有效地解决了直接数字控制方式计算机多回路集中控制带来的危险集中的问题。直至21世纪初,集散控制系统一直是世界各国包括我国在过程计算机控制方面的重要发展方向,同时,在工业生产过程控制中广泛应用集散控制系统,取得了可观的经济效益。20世纪80年代后期出现的嵌入式技术及其产品,无论是嵌入式微处理器、嵌入式组件,或是嵌入式操作系统等,都已在计算机控制系统中获得了广泛应用。

20世纪90年代,基于现场总线技术的数字变送器出现,它们之间可用现场通信网络互联、统一组态构成控制回路,新一代的分布式网络集成的**现场总线控制系统**(FCS)逐渐形成,并在近几年得到推广应用。微型计算机控制技术还迅速渗透到机电控

制领域，不仅在航空航天、军事装备、机器人、机械生产自动线中得到广泛应用，而且在家庭生活设施中，如洗衣机、微波炉、空调机等，都有由微型计算机构成的控制系统。

计算机控制与以前的模拟调节器控制相比较，就执行功能来说都包括被控对象状态信息的检测、信息加工决策和控制信息输出这三部分。但模拟调节器的控制过程在时间上是连续并行进行的，而计算机对信息的加工是离散串行的，且计算机内也只能处理离散的数字信息。

在控制理论中，计算机控制系统属于**离散控制系统**，它是指时间上离散或采用断续控制方式的一类控制系统，也称离散时间控制系统或采样控制系统。它的特点是系统内的信号仅在特定的离散瞬时表现为时间的函数。在自动控制技术的发展初期，一些控制系统由于采用了断续工作的机械式测量或控制元件，导致离散控制系统理论的产生。数字计算机被引入控制系统后，由于其处理的信息在时间上是离散的，因此在系统设计时，如果系统中有连续的部分，则需将它们离散化处理。

离散控制系统中的被控参数测量、控制算法的计算及控制信号输出均是在离散的时间点上进行的。测量信号的离散化过程称为采样过程，一般是以固定的周期（称为采样周期）在离散的瞬间把连续信号转换成脉冲序列。为了理论计算和数据处理的方便，需要在每一次采样之后使采样输出信号保持不变。在计算机控制中，测量信号的采样过程由采样器和**模数转换**（A/D）**器**实现。图 1 给出了信号的采样过程，其中 $f(t)$ 为原始的测量信号，$f_s(t)$ 为对 $f(t)$ 采样的脉冲序列，$f_h(t)$ 为对 $f_s(t)$ 作保持处理后最终的阶梯型采样输出信号，T_s 为采样周期。采样周期的选择对于系统性能有极大的影响。选择采样周期的理论依据是香农（Shannon）采样定理。该定理指出：采样频率（即采样周期 T_s 的倒数）必须大于模拟信号 $f(t)$ 频谱中最高频率 f_{max} 的两倍，才能从采样输出信号 $f_h(t)$ 中复原 $f(t)$。但在实际应用中，由于测量信号的最大谐波频率难以确定，因此，采样周期的选择大多通过试验或经验来确定。

为了适应离散控制系统中被控参数的这种采样机制，在控制策略的设计和计算中需采用离散时间数学模型。基本的离散时间数学模型是差分方程，它表示了在相邻的离散瞬间系统中各变量之间的关系。对于线性离散控制系统差分方程模型，还可以通过 Z 变换法将其转换为代数方程，使离散控制系统的分析和设计工作得到简化。

基于被控对象的离散时间数学模型设计出来的控制规律，通常也表示为在离散瞬间的时间函数。与测量信号的采样过程类似，这些在离散时间点上的控制作用也需通过保持处理，变换为阶梯型的控制信号后作用于被控对象。在计算机控制系统中，这一过程是由**数模转换**（D/A）**器**完成的。

与连续控制系统相比，离散控制系统具有数字系统的一些独特优点，如：①在很多场合，其结构比连续系统简单；②信号的传递和转换有较高的精度；③抗干扰性能较好；④适于采用数字计算机作为控制器，可以一定的精度高速完成复杂的计算任务，易实现对于复杂被控对象的高级控制策略，例如模型预测控制、多变量控制、实时优化控制等。

经典的计算机控制通常由控制部分和被控对象构成。其控制部分除了图 2 所示的硬件部分，还包括软件部分。计算机控制软件包括系统软件和应用软件。系统软件一般包括**操作系统**、语言处理程序和服务性程序等，它们通常由计算机制造厂为用户配套，有一定的通用性。应用软件是为实现特定控制目的而编制的专用程序，如数据采集程序、控制决策程序、输出处理程序和报警处理程序等。它们涉及被控对象自身的特征和控制策略，也涉及控制部分中相应部件的特性等，由实施控制系统的专业人

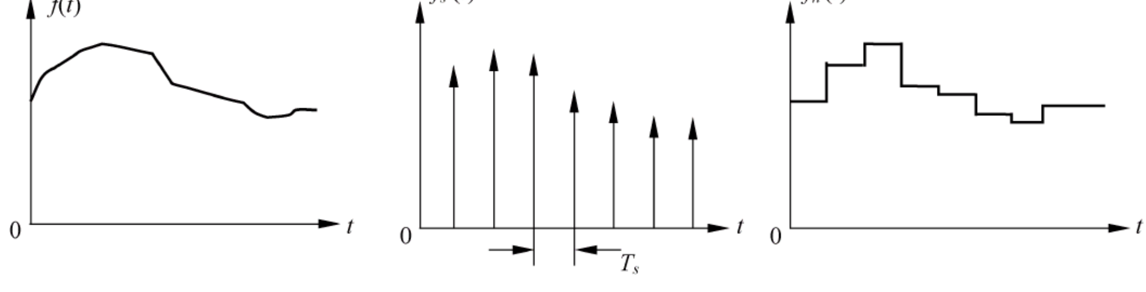


图 1　采样过程示意图

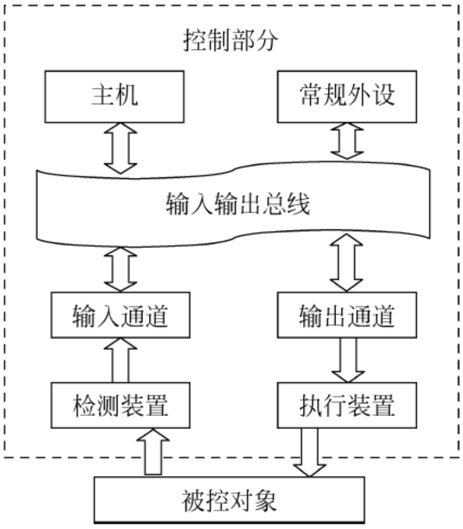


图 2　经典计算机控制系统组成框图

员自行编制或组态。对于不同规模的计算机控制系统,例如打包机的单片机控制系统和大型炼油生产装置的集散控制系统,硬件和软件部分都可以差异很大。

由于使用计算机作为控制工具,使控制系统及控制技术在深度和广度上都在不断发展。在广度上,向着大系统或系统工程方向发展。从对单一过程、单一对象的局部控制,发展到对整个工厂、企业,甚至对社会经济、国土利用、生态平衡等大规模复杂对象进行控制,实现控制和管理一体化。在深度方面,则向着智能化发展,不仅引入自适应、模糊决策、神经网络、自学习等控制方法,而且模拟人的视觉、听觉和触觉,能识别文字、图像、言语、物体,根据感知的信息进行直观判断、推理分析、自行学习解决问题,获得更高层次的控制效果。

此外,随着网络技术、现场总线技术、无线通信技术以无线传感器技术的发展,计算机控制已朝着以网络通信技术为核心的计算机控制体系结构,例如远程移动控制,现场总线控制等方向发展。

参考文献

1. 王锦标. 计算机控制系统. 2 版. 北京:清华大学出版社,2008

2. 冯培悌. 计算机控制技术. 杭州:浙江大学出版社,1990

3. Love J. Process automation handbook: a guide to theory and practice. Springer, 2007　　(王慧)

jisuanji leixing

计算机类型(computer category)　按某个或某些特征将计算机分成的若干种不同的类型。不同的分类方法使用不同的分类特征,得出不同的计算机类型。

计算机分类的方法和所分成的类型是随着计算机技术和应用的进展而不断变化的。早在 20 世纪 50 年代,已开始对电子计算机进行分类,当时把计算机按应用对象划分为科学计算用计算机和商业数据处理用计算机两类。前者采用二进制,有浮点运算,字长固定;后者用定点十进制运算,字长可变。60 年代中期,由于出现了通用系列计算机,使上述两类计算机的界限消失,计算机开始按性能高低和规模大小分类。

按性能和规模分类

从 20 世纪 60 年代中期开始的按计算机性能高低和规模大小进行的分类,到 80 年代初最终形成了把计算机分为巨型计算机(包括**小巨型计算机**)、大型计算机、中型计算机、小型计算机(包括超级小型计算机)、工作站和个人计算机等几种类型的格局。

大型计算机是通用系列计算机中的高端机种,它采用所在时期的最先进技术,性能高、容量大。在一个通用计算机系列中还有中档和低端机种,它们的性能和规模逐级降低,可以分别归入中型计算机和小型计算机类型。同一个系列中的各档计算机采用相同的操作系统,可以互相兼容(向上兼容和向后兼容)。

专门设计的**小型计算机**也出现于 20 世纪 60 年代中期。它的字长短、容量小,初期产品的指令系统比较简单。在 70 年代先后出现了系列化的小型系列计算机和超级小型系列计算机。超级小型计算机(参见**小型计算机**)性能介于小型计算机和大中型计算机之间,字长一般为 32 位,有的达到 64 位。

巨型计算机(或称超级计算机)出现于 20 世纪 70 年代,它比大型计算机的速度更快,性能更高,是专为科学工程领域大量数值计算和数据处理的需要而研制的。最初有两种结构:单指令流多数据流(SIMD)阵列处理机(参见**阵列处理机**)和流水线向量计算机(参见**向量计算**)。到 80 年代又出现了多指令流多数据流(MIMD)的大规模并行处理(MPP)系统(参见**大规模并行处理**)。80 年代中期出现了小巨型计算机,它的价格与超级小型计算机相当,但性能接近巨型计算机。最初的小巨型计算机采用巨型计算机的高速流水线技术,但在指令系统、数据结构和存储管理等方面较多地吸收超级小型计算机的特点,使之有较高的性能价格比。其后又出现用微

处理器并行处理系统的小巨型计算机,由于与相邻的类型在性能、规模和结构上的界限日趋模糊,小巨型计算机这一名词到90年代已不再使用。

自从20世纪70年代初出现**微处理器**之后,在计算机类型的最低端又添加了**微型计算机**。80年代初,微型计算机分为**个人计算机**和**工作站**两种类型。至此,在80年代形成了巨型计算机(包括小巨型计算机)、大型计算机、中型计算机、小型计算机(包括超级小型计算机)、工作站、个人计算机的分类格局。但随着微电子技术的发展,巨型计算机和大型计算机系统结构的先进技术不断下移到低端机中,而低端的机种,特别是工作站和个人计算机的性能更是不断提高,使上述分类中的各相邻类型间的界限越来越模糊。

进入90年代后,计算机的分类类型发生了变化。原来采用双极型电路的巨型计算机、大型计算机乃至小型计算机都先后采用由CMOS微处理器组成的多处理机结构,原有的机种逐渐消失,新出现的一些计算机仅仅是在并行处理机的数量、采用的互连技术和微处理器型号等方面有所不同。与此同时计算机在网络和通信中的地位日显突出,因而在商品市场上大量销售的计算机类型逐渐为**服务器**(包括超级服务器)、工作站、台式个人计算机和**移动式计算机**这样一些与网络和通信相关的机型所代替。这一时期,还把性能和规模处于最高端的计算机称为高性能计算机(参见**高性能计算**)。高性能计算机一般可以用来做成超级服务器,但某些性能极高和规模极大的高性能计算机往往专门用于解决某方面极为复杂的问题,例如日本专用于地球和气候模拟的地球模拟器、美国专用于蛋白质结构分析的Blue Gene系统以及一些专用于核模拟的超大规模并行机。

服务器是在网络环境或客户-服务器环境中为客户提供服务的计算机。服务器的性能和规模涵盖了很大的范围,可把服务器由大到小分为企业级服务器、部门级服务器和群组级服务器。企业级服务器是服务器中的高端,相当于原来的大型计算机。每一种服务器系列都有良好的可伸缩性,在有的服务器系列产品中,并行工作的处理机可从4台一直扩充到几百台甚至于上千台。极高性能的服务器称为超级服务器。服务器的结构有分布式共享存储器的多处理机系统(参见**共享存储**)、松耦合的大规模并行处理(MPP)多计算机系统(参见**大规模并行处理**)或集群式计算系统(参见**集群计算**)。很多服务器设计成专供某一方面的应用,例如Web服务器、邮件服务器、文件服务器、数据服务器等;有的则设计成多功能服务器,可以支持科学与工程计算、数据处理、事务处理和网络信息服务等多方面的应用。通过虚拟化技术,可以在同一套硬件平台上同时运行多个不同的操作系统和相应的服务软件,并根据不用应用的实际需求,动态调整分配给各个服务的硬件资源[如中央处理器(CPU)数、内存容量、磁盘容量、网络带宽等],实现更高效能的计算。

工作站属于**台式计算机**,其性能高于个人计算机,它可以在个人计算环境中或在分布式网络计算环境中工作,主要用于**计算机辅助设计**、制图等领域。一般由1到4个高性能精简指令集微处理器芯片组成,采用对称式共享存储器结构。

台式个人计算机一般是指单个用户独用的价格低廉的**微型计算机**,用于联网发送电子邮件、网络浏览、文字处理、游戏等。它通常采用单个x86微处理器芯片(早期为单核芯片,近年已发展到多核芯片)。随着x86微处理器性能的不断提升,个人计算机与工作站间的界限也逐渐模糊。有时,人们把多CPU、大内存、高图形配置的个人计算机亦称为工作站。

移动式计算机发展迅速,已开发出笔记本计算机、手持计算机和可穿戴计算机等。移动式计算机一般可以通过无线传输成为移动的结点,实现移动网络计算。**笔记本计算机**可以随身携带,它开始出现于20世纪80年代末,现在在许多场合已经可以替代台式个人计算机。早期的手持计算机是一种**个人数字助理**,可以拿在手里使用。近年流行的平板也是一种手持计算机,它用方便灵活的触摸屏替代了键盘和鼠标,使用更为方便。可穿戴计算机(参见**可穿戴计算**)的部件被设计成一些可以合理分布于人体多个部位的模块,各个模块间的互连由近距离无线通信实现。它具有十分友好的人机交互能力,可以将人的双手从键盘和鼠标上解放出来。

从系统结构角度分类

在计算机学术界内,有人从计算机系统结构的角度对计算机进行分类。最为流行的是1966年M J Flynn提出的分类法,这是按指令流和数据流的多重性,把计算机划分为单指令流单数据流(SISD),单指令流多数据流(SIMD),多指令流单数据流(MISD)和多指令流多数据流(MIMD)4种类型(参见**并行处理系统**)。在已有的并行处理系统中,除个别的属于SIMD外,其余的全部都属于MIMD

类型。根据互连网络和通信机制的不同,MIMD 可分为紧耦合的共享存储器多处理机系统和松耦合的多计算机系统两大类。共享存储器多处理机系统又可根据共享存储器的物理位置,分为集中式共享存储器多处理机系统和分布式共享存储器多处理机系统。共享存储器多处理机还可根据每个处理机的作用是否完全相同分为对称式多处理机(SMP)系统和主从式多处理机系统两类。松耦合的多计算机系统可根据互连网络和通信机制的不同,分为消息传递多计算机系统(参见**消息传递**)和集群式计算机系统(参见**集群计算**)。多处理机系统还可以按所用的处理机是否相同分为两类:同构型多处理机系统和异构型多处理机系统。

另外,根据数据和指令在存储器中存放方式的不同,可以把计算机分为冯·诺依曼结构和哈佛结构两类。冯·诺依曼结构的计算机中,数据和指令统一以二进制形式存放在同一个存储器中,数据和指令没有明确的区分。哈佛结构的计算机中,指令和数据存放在各自独立的存储器中,分别称为指令存储器和数据存储器。在实际的计算机系统中,冯·诺依曼结构采用得较多,但大部分数字信号处理机都采用哈佛结构。有时,把采用分离的指令高速缓冲存储器和数据高速缓冲存储器的计算机,也称为哈佛结构,尽管其主存储器中,数据和指令仍然是混放的。

可以按计算机指令系统的复杂程度把计算机分类为**精简指令集计算机**、**复杂指令集计算机**。还可按计算机对高级语言支持的程度,或者说按机器语言与高级语言的差距,把计算机分为和语言对应的系统结构的计算机和直接执行语言的系统结构的计算机。在和语言对应的系统结构的计算机中,机器语言与高级语言基本上是一一对应的,但需要经过软件或硬件翻译。直接执行语言的系统结构的计算机能直接执行高级语言原码,不需要经过中间翻译。

还可以按计算机驱动原理的不同把计算机分为控制驱动、数据驱动和需求驱动三种类型的计算机。控制驱动的计算机就是传统的以冯·诺依曼结构为基础的计算机,它执行操作的次序受指令计数器的控制。后两种是非传统计算机,数据驱动的计算机就是**数据流计算机**,它的操作在全部操作数都到齐时即开始执行;需求驱动的计算机就是**归约机**,它的操作仅在需要用到这个操作的结果时才开始执行。

按其他方法分类

在数字计算机发展的过程中,曾以电子开关器件的更新作为计算机分类的特征,把电子管计算机称为第一代计算机,晶体管计算机称为第二代计算机,中小规模集成电路计算机称为第三代计算机,20 世纪 70 年代中期以来采用大规模和超大规模集成电路的计算机统称为第四代计算机。

计算机还可以从应用范围的角度分为通用计算机和专用计算机。通用计算机可用于各种不同类型的应用,例如科学与工程计算、数据处理、事务处理和过程控制等。专用计算机只适合某一方面特殊的应用,例如专门用于控制生产过程的**过程控制计算机**。随着微电子技术的发展,通用微处理机芯片的集成度和性能价格比不断提高。在很多场合下通用计算机已可涵盖和替代专用计算机。在另外很多场合下,专用计算机可以直接装入机电设备、仪器仪表或家电设备内部,成为其中的一个部件,这就是**嵌入式计算机**。

在计算机采用双极型线路的年代里,还有一种位片计算机。当时芯片的集成度低,有一处做法是把计算机的各种逻辑功能部件沿字长方向划分为一定宽度的标准模块。把这些模块级联起来,就可以组成任意宽度的数据通路,从而构成一台计算机,这就是位片计算机。进入 20 世纪 90 年代以后,位片计算机已经不再生产。还有的把中央处理机、存储器和输入输出接口集成在一个芯片上的微型计算机称为**单片计算机**或单片机,主要用于嵌入式控制领域。

对于在可靠性要求很高的场合下使用的计算机,如果计算机在出现故障时没有或几乎没有停顿时间,计算机仍能继续运行,这种计算机就是**容错计算机**。如果出现故障时还需要有几秒钟、几分钟乃至更长的间断时间来采取自动弥补措施,则称为**高可用计算机**。还有为适应军事、航天、野外、海洋、工业等特定恶劣环境而设计的**抗恶劣环境计算机**,为满足环境保护要求而设计的**绿色计算机**等。

除了用微电子技术做成的集成电路组成计算机外,还在研究传统的电子器件以外的基本部件和工作原理。例如用有机或生物材料构成的分子器件、光器件和量子器件,分别用它们组成具有特色的**生物计算机**、**光计算机**和**量子计算机**。

(孙强南　唐志敏)

jisuanji liushuixian

计算机流水线(computer pipeline)　把计算机的指令或费时的复杂操作分解成一系列可以独立

执行的简单步骤,按流水线方式交迭进行的一种操作方式。每个步骤的执行部件称为站,每个站将本步的执行结果送往下一站后,又去执行下一条指令或下一个操作的这一步。这类似于工厂的生产流水线,工件放在传送带上,顺序地流过各个工位,每个工位在规定的时间内,完成本工位的装配任务。流水线的特点是在一项任务未完成前,可以启动新的任务,任务完成的速率与任务流过流水线的总时间无关,仅取决于任务送入流水线的速率,从而提高了整体处理速度。

计算机流水线的站由组合逻辑线路组成,完成特定的算术和逻辑操作。相邻的两个站由寄存器隔开,信息在站间的流动由时钟信号 C 来实现同步。图 1 是流水线基本结构的示意图。在图中,L 表示寄存器。

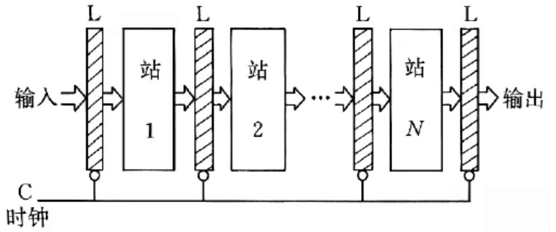


图 1 流水线基本结构示意图

将流水线概念引入计算机,可以追溯到 1946 年冯·诺依曼等人设计的第一台存储程序计算机。当时,提出了在运算器和输入输出设备之间设置一个缓冲器,使运算和输入输出操作可以同时进行。尽管冯·诺依曼在他的第一台计算机中并没有实现这种技术,但是,在后来的计算机中,这种运算和输入输出操作的重叠概念被广泛采用,并且成为计算机流水线的原始形式。

在晶体管计算机时代,计算机的主存储器采用磁芯。由于磁芯存储器的存取周期比晶体管中央处理器的周期长 10 倍左右,因此,中央处理器在执行指令的同时,可以启动 1 个或多个并发的存储器访问,这就构成了访存流水线。

计算机流水线设置多个站来提高处理速率。在 20 世纪 60 年代,逻辑元件的价格相对较高,流水线技术只用于价格昂贵的大型或巨型计算机。60 年代初,IBM 公司的 STRETCH 和 CDC 公司的 6600 计算机是典型的大量使用流水线技术的计算机系统,对后来的大型和巨型计算机的体系结构产生了深刻的影响。随着微电子技术的迅猛发展,到 80 年代,逻辑元件的价格、功耗都大幅度下降,靠增加硬件来获得高速度的流水线技术在各种档次、各种类型的计算机中都得到了广泛应用。甚至在价格十分低廉的微处理器中,也能在执行运算的同时,使用流水线技术访问存储器。

计算机流水线工作原理

流水线的加速原理可用图 2 来说明。图 2(a)表示串行处理过程,假如 1 个作业有 N 步,完成 1 步需要 1 个单位时间,则完成 N 步需要 N 个单位时间。图中的每一个方框表示执行 1 步,方框中的标号为执行的步号。图 2(b)表示 M 个作业用流水线技术处理的情况。图中每一行方框表示 1 个作业随时间的变化,每一列方框表示在某个特定时刻,作业流在流水线中执行的步号。可看出,在图 2(a)中,完成 1 个作业需要 N 个周期,而在图 2(b)中,如果 $M \gg N$,则完成 1 个作业只需约 1 个周期。

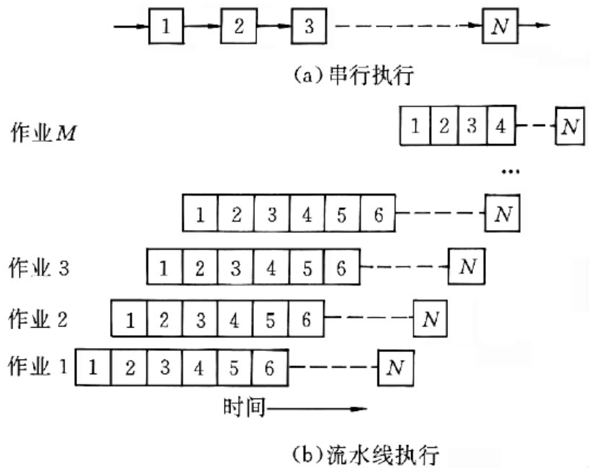


图 2 流水线加速原理

计算机流水线的分类

计算机流水线一般可以分为指令流水线和运算流水线。

1. 指令流水线

典型的指令执行过程可分解为若干步骤,这里假定为 5 步:①取指令;②指令译码;③形成操作数有效地址;④取操作数;⑤执行指令,含存操作结果和修改程序计数器。假如把这 5 个步骤分别安排在相互独立工作的站中进行,并且使每个站的执行时间大致相同,都是 1 个时间单位。这 5 个步骤的处理过程可按图 2(b)实现指令流水线。

在一般情况下,流水线的第一站要连续不断地取指令,通常指令是顺序存放的,可以通过程序计数器加 1 来确定下一条指令的地址。但是,遇到条件转移指令,只有到流水线的最后一站才能知道下一

条指令的地址。若在流水线的第一站不对转移类指令采取特殊措施,在一条转移指令到达最后一站产生目的地址以前可能已经有若干条本不该执行的指令进入了流水线,改变了 1 个或多个机器寄存器的状态。为了保证正确执行,最简单的方法是使取指令和执行指令这两步进行互锁。在第一站取出条件转移指令后,直到它到达最后一站前都不取新的指令,在最后一站产生正确的目的地址后才解除互锁,并使第一站用新的地址继续取指令。采用互锁时,流水线遇到转移指令就形成一段空闲状态。在大多数计算机中,条件转移出现的频率很高,平均 5 ~ 10 条指令出现 1 次,简单的互锁将损失很多时间。

为了解决转移指令对流水线效率的影响,提出了多种方案。对无条件转移指令提出了设置指令缓冲区的方法。在指令缓冲区中,预取足够多的指令,假如无条件转移后的地址落在指令缓冲区中,无条件转移可以不引起执行部件的停顿。对于条件转移指令,确定转移方向的条件码要在流水线后端的执行部件中产生,对流水线效率影响大。为了减少条件转移指令对流水线效率的影响,提出了多种方法,例如加快和提前形成条件码的方法;按一个方向猜测执行的方法;预取转移目标方法,即设置多套指令缓冲寄存器,发现条件转移指令后,同时按两个分支方向预取指令。此外,还有用编译程序自动调整条件转移指令位置的延迟转移方法等。

影响流水线效率和运行正确性的,除了转移指令外,还有数据相关。例如,在流水线中某条指令的结果可能被后面的指令使用,这时,后面的指令必须等前面指令的结果产生后才能执行。又如,流水线中某条指令的结果要存入某寄存器,但是该寄存器中原来的内容要由前面的指令来读取,那么,只有等前面指令读取该寄存器内容后,后面的指令才可写入该寄存器。再如,流水线中两条不同的指令有可能写同一地址,由于流水线会改变指令的执行次序,有可能造成该地址内容出错。

上述各种原因使指令流水线的设计变得错综复杂。在设计流水线时,必须确保计算机在执行转移指令和数据相关的情况下,仍能正确和高效地运行。

2. 运算流水线

计算机的某些运算很费时间,例如浮点运算、乘法、除法等。为加快运算速度,这些运算也可以采用流水线技术。现以浮点加法为例加以说明。

假设输入的是一对规格化的浮点数,分 4 步完成浮点加法:①对阶,找出阶码较大的数,求阶差,较小的数尾数右移;②尾数相加,产生和的尾数;③规格化和的尾数,调整和的阶码;④尾数舍入,如果舍入引起尾数溢出,再规格化和调整阶码。

上述 4 步可以构成 1 条由 4 个站组成的浮点加法流水线。

运算流水线又可以细分为单功能和多功能流水线、静态和动态流水线以及标量和向量流水线 3 类。

随着微电子技术和计算技术的发展,流水线技术已应用到从微型计算机到巨型计算机的各种类型的计算机中。值得一提的是,流水线技术已成为实现**精简指令集计算机**(RISC)的首选技术。在 RISC 中,最初的目标是每个时钟周期执行 1 条指令,后来提出了在 1 个时钟周期内执行多条指令的目标,于是便出现了超标量、超流水线和超长指令字的体系结构(参见**多发射结构**)。在每种体系结构中,出现了多条流水线或多个功能部件。此外,流水线技术应用在软件中,出现了开发循环程序指令级并行的软件流水线方法(参见**软件流水**)。总之,流水线技术已进入更高的发展阶段。

参考文献

1. Stone H S. High-performance computer architecture. 3rd ed. Reading, MA: Addison-Wesley Publishing Company, Inc., 1993

2. Hwang K, Briggs F A. Computer architecture and parallel processing. New York: McGraw-Hill, 1984

3. Hennessy J L, Patterson D A. Computer architecture: a quantitative approach. 5th ed. Morgan Kaufmann, 2011

(韩承德)

jisuanji muma

计算机木马(computer trojan)　一个有用的、或者表面上有用的程序或命令过程,但其中包含了一段隐藏的、激活时将执行某种有害功能的代码,可以用来非直接地完成一些非授权用户不能直接完成的功能。

计算机木马不自我复制,没有主动传播的功能,也不会像病毒那样刻意感染其他文件。其主要意图不是为了破坏用户的系统,而是作为一种驻留程序隐藏在系统内部,以控制为主,主要目的是为了监视并窃取系统中的有用信息、用户数据和系统控制权,因此,多不会直接对电脑产生危害。

计算机木马具有隐藏性、伪装性、精巧性、非授权性、自动运行性和自动恢复性等特性,能自动打开被植入主机的特殊端口、能执行诸如搜索特定信息、

密码口令、记录键盘操作等功能，对网络信息系统的安全、特别是信息的私密性构成严重威胁。

计算机木马的作用过程主要包括如下几个部分：①捆绑欺骗——将木马嵌入流行软件、邮件或网页中，等待植入机会；②木马植入——通过各种途径和手段将木马植入目标系统；③木马隐藏——利用程序隐蔽、进程隐蔽、API 拦截、远程线程注入、通信隐藏、协同隐藏等各种手段伪装木马程序，将木马隐藏在目标系统中，从而秘密操控远程主机，等待机会，进行破坏活动；④木马启动——通过系统配置文件、启动菜单、注册表、计划任务、控制面板、进程启动等方式启动木马，实施在高隐蔽性的状态下窃取数据信息等攻击行为。

计算机木马分为窃取信息型、窃取密码型、远程控制型、键盘记录型、分布式攻击型、邮件炸弹型、代理型、反弹端口型、程序杀手型等多种类型。

计算机木马植入技术主要有通过文件捆绑法、邮件附件法、WEB 网页挂马法等人工干预方式将木马程序安装到目标系统的被动植入技术和将木马程序通过程序自动安装到系统中的主动植入技术。

计算机木马防护需要从①防——防止木马植入到系统中；②查——通过检查系统进程、检查注册表、INI 文件和服务、检查开放端口、网络流量监控等技术手段，检查系统是否受到木马侵害，掌握木马在系统中留下的痕迹或网络通信活动特征；③阻——干扰木马正常运行，切断木马通信或中断木马进程；④杀——中断木马进程或删除木马运行所依赖的文件或配置信息等多个方面进行。具体防护模式包括终端防护和网关防护等。

计算机木马的发展趋势主要有：隐蔽性增加、植入方式多样化、编程机制多样化、跨平台性、模块化设计、更新更强的感染模式、商业病毒木马泛滥成灾。

参考文献

张仁斌，等. 计算机病毒与反病毒技术. 北京：清华大学出版社，2012

（刘宝旭）

jisuanji ruchong

计算机蠕虫（computer worm） 一种综合利用渗透、攻击、自我复制等多种技术，无须计算机使用者干预即可运行，并能够自动通过网络进行自我传播的恶意程序。它通过不停地获得网络中存在漏洞的计算机上的部分或全部控制权，自主地将自己复制到其他计算机上来进行传播。它不需要附着在其他程序上，而是独立存在的。

计算机蠕虫自身不改变其他程序，但可携带一个具有改变其他程序功能的病毒。大多数计算机蠕虫试图将其自身复制到存在漏洞的计算机上，然后使用此计算机的通信通道来进行复制。

它具有智能性、自动性、传播快速性、广扩散性、可激发性和高技术性等特征。其传播形式复杂多样，是目前危害最大的恶意软件，会极大地消耗网络资源导致大面积网络拥塞甚至瘫痪，几乎每次蠕虫发作都会造成巨大的经济损失。

计算机蠕虫的传播步骤：①搜索系统或网络，确认下一步要感染的目标；②建立与目标系统或远程主机的连接；③将自身复制到目标系统或远程主机，并尽可能激活它们。

计算机蠕虫的传染目标是互联网内的所有计算机，传播途径主要是利用操作系统和应用程序自身的漏洞主动进行攻击，还可利用共享文件夹、恶意网页和电子邮件等进行复制和传播。

计算机蠕虫以计算机为载体，以网络为攻击对象，计算机病毒需要在计算机的硬盘或文件系统中繁殖，而计算机蠕虫只会在内存中维持一个活动副本，甚至根本不向硬盘中写入任何信息。网络的发展使得蠕虫病毒可以在几个小时内蔓延全球，其破坏力和传染性不容忽视。

计算机蠕虫一旦在系统中激活，就可以向系统注入特洛伊木马程序，或者对被注入计算机进行任何次数的破坏行动。

计算机蠕虫分为传统蠕虫、邮件蠕虫和漏洞蠕虫等几种类型，其中，漏洞蠕虫是发展最快，也是最常见的蠕虫类型，如冲击波、震荡波、红色代码、SQL 蠕虫王等。

计算机蠕虫的防护技术主要有监测与预警、传播抑制、仿生、漏洞检测与系统加固、免疫、阻断与隔离、清除等。

计算机蠕虫防护策略：①防——防止计算机蠕虫侵入到网络系统，特别是防止大规模爆发；②查——检查网络系统是否受到计算机蠕虫侵害，掌握计算机蠕虫在系统留下痕迹或网络通信的活动特征；③阻——干扰计算机蠕虫正常运行，切断计算机蠕虫的传播或中断计算机蠕虫程序运行；④杀——在掌握计算机蠕虫特征的基础上，中断计算机蠕虫的进程或删除计算机蠕虫运行所依赖的文件或配置信息。

参考文献

张仁斌，等. 计算机病毒与反病毒技术. 北京：

清华大学出版社，2012　　(刘宝旭)

jisuanji ruanjian

计算机软件(computer software)　计算机系统中的程序及其文档。**程序**是计算任务的处理对象和处理规则的描述，文档主要是为了便于了解程序所需的阐明性资料。程序在计算机平台之上运行，文档一般是给人看的；软件通常通过解决用户的问题而表现出其价值。

细言之，软件一词具有三层含义。一为个体含义，即指计算机系统中的单个程序及其文档；二为整体含义，即指在特定计算机系统中所有上述个体含义下的软件的总体；三为学科含义，即指在研究、开发、维护以及使用前述含义下的软件所涉及的理论、原则、方法、技术所构成的学科。在这种含义下，软件宜称为软件学，但一般仍称作软件。

软件一词源于程序，到了20世纪60年代初期，人们逐渐认识到和程序有关的文档的重要性，从而出现了软件一词。

软件是用户与硬件之间的接口，用于解决用户计算的问题。要使用计算机，就必须编制程序，必须有软件。用户主要是通过软件与计算机进行交往。软件是计算机系统设计的重要依据。为了方便用户，为了使计算机系统具有较高的总体效用，在设计计算机系统时，必须通盘考虑软件与硬件的结合以及用户的要求和软件的要求。软件在计算机系统中起指挥、管理作用。计算机系统工作与否，做什么以及如何做，都是听命于软件。

发展计算机科学技术，软件、硬件和应用都是不可缺少的重要方面。三者既有分工，又有配合。软件的发展以硬件为基础，应用为目的，其发展也促进了硬件、计算机应用技术以及其他科学技术的发展。它在社会信息化和人类文化的发展中具有重要作用。

发展过程

软件的发展受到应用和硬件发展的推动和制约，其发展过程大致可分为三个阶段。

从第一台计算机上的第一个程序的出现到实用的高级程序设计语言出现以前为第一阶段(1946～1956年)。计算机的工作是由储存在其内部的程序指挥的。这是冯·诺依曼式计算机的重要特色。当时计算机的应用领域较窄，主要是科学计算。就一项计算任务而言，输入、输出量并不大，但计算量却较大，主要处理一些数值数据。机器结构以中央处理器为中心，存储容量较小。编制程序(简称编程)所用的工具是低级语言，即以机器基本指令集为主的机器语言和在机器语言基础上稍加符号化的汇编语言。突出的问题是，程序的设计和编制工作复杂、烦琐、费时和易出差错。衡量程序质量的标准主要是功效，即运行时间省、占用内存小，很少考虑到结构清晰、易读性和易维护性。设计和编制程序采用个体工作方式，强调编程技巧，主要研究科学计算程序、服务性程序和程序库，研究对象是顺序程序。当时人们对和程序有关的文档的重要性尚认识不足，重点考虑程序本身，尚未出现软件一词，但毕竟由于程序是软件的主体，从发展的连续性来看，故将其归为第一阶段。

从实用的高级程序设计语言出现以后到软件工程出现以前为第二阶段(1956—1968年)。随着计算机应用领域的逐步扩大，除了科学计算继续发展以外，出现了大量的数据处理问题，其性质和科学计算有明显区别。就一项计算任务而言，计算量不大，但输入、输出量却较大。这时，机器结构转向以存储控制为中心。出现了大容量的存储器，外围设备发展迅速。为了提高程序人员的工作效率，出现了实用的高级程序设计语言。为了充分利用系统资源，出现了操作系统。为了适应大量数据处理问题的需要，开始出现数据库及其管理系统。在20世纪50年代后期，人们逐渐认识到和程序有关的文档的重要性，从而到了60年代初期，出现了软件一词，融程序及其有关的文档为一体。这时，软件的复杂程度迅速提高，研制周期变长，正确性难以保证，可靠性问题相当突出。到了60年代中期，出现了人们难以控制的局面，即所谓软件危机。为了克服这一危机，人们进行了以下三个方面的工作：①提出结构程序设计方法；②提出用工程方法开发软件；③从理论上探讨程序正确性和软件可靠性问题。这一阶段的研究对象增加了并发程序，着重研究高级程序设计语言、编译程序、操作系统以及各种应用软件。计算机系统的处理能力得到加强，设计与编制程序的工作方式逐步转向合作方式。

软件工程出现以后迄今为第三阶段(1968年以来)。由于大型软件的开发是一项工程性任务，采用个体或合作方式不仅效率低，产品可靠性差，甚至很难完成，只有采用工程方法才能适应。从而在1968年的大西洋公约学术会议上提出了软件工程。40多年来，软件领域工作的主要特点是，第一，随着微型化、并行化、网络化硬件平台的快速发展，资源

化、服务化、泛在化应用形态的不断拓广，各类新型应用模式与应用系统不断出现，如普适计算、服务计算、云计算、嵌入式系统、混成系统、信息物理融合系统等，对计算机软件提出了挑战，出现了嵌入式软件与应用、网络软件与应用、分布式软件与应用、自治式软件与应用、“软件作为服务”与应用等各种软件与应用形态，促进了软件理论、方法与技术的发展。第二，开发方式逐步由个体合作方式转向工程方式，软件工程发展迅速，出现了计算机辅助软件工程、软件开发过程的规范化管理、基于 Internet 的软件协同开发等。除了开发各类工具与环境，用以支撑软件的开发与维护外，还出现了一些实验性的软件自动化系统。第三，软件复杂性控制问题颇受关注，致力研究软件开发过程本身，在主流的功能分解风范与模型、面向对象风范与模型基础上，开始探索新型软件开发风范与模型，如资源开放联盟、网构软件风范与模型等。研究软件体系结构、基于构件的软件以及中间件等。第四，除了软件传统技术继续发展外，人们着重研究以智能化、自动化、集成化、网络化、并行化以及自然化等为标志的软件开发与运行新技术，开始探索情景驱动的软件自适应方法、技术与系统等。第五，注意研究软件理论与非经典的软件模型，特别是软件开发过程的本质、量子程序语言与系统等。

分　类

软件一般可分为系统软件、支撑软件、应用软件三类。

系统软件　居于计算机系统中最靠近硬件的一层。其他软件一般都通过系统软件发挥作用。它与具体的应用领域无关，如编译程序和操作系统等。编译程序将程序人员用高级语言书写的程序翻译成与之等价的、可执行的低级语言程序；操作系统则负责管理系统的各种资源、控制程序的执行。在任何计算机系统的设计中，系统软件都要予以优先考虑。

支撑软件　支撑软件的开发、维护与运行的软件。随着计算机科学技术的发展，软件的开发、维护与运行的代价在整个计算机系统中所占的比重很大，远远超过硬件。因此，支撑软件的研究具有重要意义，直接促进软件的发展。当然，数据库管理系统、网络软件等也可算作支撑软件。但是 20 世纪 70 年代中后期发展起来的软件开发环境以及后来开发的中间件则可看成现代支撑软件的代表，软件开发环境主要包括环境数据库、各种接口软件和工具组。三者形成整体，协同支撑软件的开发与维护。

应用软件　特定应用领域中用以解决实际计算问题的软件。例如，人口普查用的软件就是一种应用软件。对于具体的应用领域，应用软件的质量往往成为影响计算机实际效果的决定性因素。20 世纪 70 年代出现的嵌入式应用与近年来兴起的信息物理融合系统，其相应软件的复杂程度高，开发工作量大，促进了软件的发展。模拟应用导致模拟语言(SIMULA)的出现。随着计算机应用水平的不断提高，各类应用模式不断出现，促进了计算机软件的发展。应用软件的作用越来越大。

基本内容

主要有软件语言、软件方法学、软件工程以及软件系统等。

软件语言　用以书写软件的语言。从刻画程序的级别看，可分为需求级语言、功能级语言、设计级语言、实现级语言以及文档语言。需求级语言用以书写软件需求定义，又称需求定义语言，目前多数仍是非形式的语言或半形式的语言；功能级语言用以书写软件功能规约，又称功能规约语言，可以是形式的，也可以是非形式的；设计级语言用以书写软件设计规约，又称设计规约语言，一般是形式的；实现级语言用以书写实现算法，相对说来，较为成熟，低级语言、高级语言从 FORTRAN，ALGOL，COBOL，PASCAL，C，直到 Ada 以及面向对象的 C + + 、Java 等均属此类；文档语言用以书写文档，目前一般是非形式的。从语言风范看，可分为命令式语言与申述式语言；从语言的应用范围看，有通用语言与专用语言。当然，还有其他分法，详见“**程序设计语言**”。

软件方法学　软件开发全过程的指导原则与方法体系。其另一种含义是以软件方法为研究对象的学科。从开发风范上看，软件方法有自顶向下的开发方法、自底向上的开发方法。在实际软件开发中，大都是自顶向下与自底向上两种方法的结合，只不过是以何者为主而已。从性质上看，有形式方法与非形式方法。形式方法是一种具有坚实数学基础的方法，从而允许对系统和开发过程作严格处理和论证。非形式方法则不把严格性作为其主要着眼点。从适用范围来看，有整体性方法与局部性方法，适用于软件开发全过程的是整体性方法，自顶向下方法、自底向上方法、各种软件自动化方法等均为整体性方法。适用于开发过程个别阶段的为局部性方法，如适用于需求分析阶段的各种需求分析方法，适用于设计阶段的各种设计方法等。从适应能力看，有敏捷性方法和非敏捷性方法。敏捷性方法是以目标

用户的深度参与、开发者之间的高效沟通、系统的迭代开发等手段来敏捷适应变化的轻量级方法，而非敏捷性方法更加强调以严谨的计划、规范的文档和预定的过程来提高软件开发过程的可预测性和可管理性。此外，由于程序设计方法的发展相对较为成熟，从而早在软件方法学出现以前，就出现了程序设计方法学，它研究各类程序设计方法，如过程式程序设计、逻辑式程序设计、函数式程序设计、对象式（面向对象）程序设计以及顺序程序设计、并发程序设计、并行程序设计、分布程序设计、可视程序设计、文化程序设计等。

软件工程　应用计算机科学、数学及管理科学等原理，以工程化方法制作软件的工程，它是一门交叉性学科。软件工程的架构可用一三元组刻画，即 SE =（G，P，Q），其中 SE 表示软件工程，G 为目标，P 为原则，Q 为过程。目标 G 主要包括正确性、易用性以及价格合宜，正确性反映软件产品实现相应功能规约的程度，E. W. Dijkstra 曾说，“迄今的软件排错设施只能发现软件错误，不能断定软件没有错误”，20 世纪 60 年代中期以来，虽有不少计算机科学家致力于正确性问题的研究，但迄今为止，不论理论上还是实践上正确性问题均未得到很好解决；易用性反映软件的基本结构、实现及其文档为用户易用的程度，由于软件系统的用户广泛，要求因人而异，因而，易用性往往是软件工程难以满足的目标；价格合宜反映软件开发与运行的总代价满足用户要求的程度，随着硬件价格的日趋低廉，传统的系统性能显得不太重要，价格合宜扩大成包括软件运行与维护方面的价格考虑，由于复杂软件系统不可能一步开发成功，通常采用增殖式开发策略，先生产出一个基本核心产品，随后再改进提高，以实现所需的全部功能。总体价格合宜便要求产品能便于修改与改进。原则 P 主要包括易变性原则、综合性原则、支撑性原则以及管理性原则等，其中易变性原则指必须认识软件需求的易变性；综合性原则指综合使用各种方法与工具以实现软件目标；支撑性原则指应充分认识支撑软件的作用；管理性原则指必须认识管理过程的作用。过程 Q 涉及软件生存周期中的各类过程，主要包括基本过程、支撑过程以及组织过程。

软件系统　计算机系统中由软件组成的子系统，其组成成分可有系统软件、支撑软件以及应用软件。系统软件与特定解题领域无关，它主要包括操作系统、语言处理系统、数据库系统、分布式软件系统、网络软件系统以及人机交互软件系统等。支撑软件主要包括用以支撑软件开发、维护与运行的各类软件开发环境和中间件系统。应用软件是特定应用领域专用的软件，种类繁多，不拟枚举。

此外，随着软件在人类生产生活中的作用日益突出，与软件开发和使用相关的法律、社会和心理问题亦成为重要的研究内容。

难点与展望

软件是智力产品，软件开发是智力活动。其难点是：第一，如何从软件科学的角度加深对软件开发过程本质的认识。“变是不变的真理”在软件领域表现突出。软件在使用过程中必须不断维护，其中包括：①纠错性维护，使用中一旦发现存在错误，便及时纠正；②预见性维护，指预防检测与改正软件中存在的潜在错误；③适应性维护，随着环境的改变，使原有软件适应新的环境；④完善性维护，原有软件在设计时难免有不完善之处，在使用中视需要逐步完善。变动性是软件的重要特征。软件开发风范与模型均和领域有关，领域不同，合适的风范与模型可能也不同。但软件开发过程的本质是否也和领域有关，目前尚难断定。共性为何，个性为何，这些都是有待研究的问题。软件主要是由人开发的。各人思考问题的习惯、方式不尽相同，思维规律也不尽相同，但软件开发过程的本质是否也和思维规律有关，亦亟待研究。此外，软件开发过程的复杂性研究对软件开发的作用巨大，影响深远。第二，如何从开发与运行的角度，使得计算机软件能够更加适应开放环境的需要，即功能与性能更加智能化。“智能”含义，众说纷纭。然而，多数人认为，智能是指洞察、学习、理解、推理的能力，其中学习是核心。如果软件智能化指的是使软件本身具备智能，则事实表明，软件是能智能化的。如果软件智能化指的是使软件具备人类智能，则当然是有条件的。目前流行的归纳学习、分析学习、联接学习以及遗传学习等途径各有利弊，应视具体情况，酌情采用，或另创新途径。另一方面，需要从软件方法学的角度，提出新的软件范型、结构模型与相应的开发方法与运行机理，对智能化的软件给以系统化的支持。第三，如何从保证软件产品质量的角度来保证程序正确性与软件可信性。程序正确性总是相对某一基准而言的。一般说来，程序（实现级）是正确的，指的是它能全部体现相应设计规约中的功能；设计规约的正确性，指的是它能体现功能规约中的全部功能；功能级程序（即功能规约）是正确的，则指它能体现需求定义中的全部需求。过去考虑较多的是实现级程序的正确

性，今后宜更多考虑设计级与功能级程序的正确性。理论上的正确性与实际的正确性从理论上说应该是一致的，但实际上又往往不能完全一致。问题在哪里，是理论原因还是实际原因，有待研究。此外，开发过程不正确，则任何级别的程序正确性均无法保证。在这种意义上说，开发过程的正确性就显得格外重要。许多开放环境下的复杂软件系统的正确性没有绝对的基准，而只能以用户满意度或"足够正确"来刻画。如何处理此类系统的正确性问题是亟待研究的问题。更进一步，如何保证开放环境下软件系统高可信也是一个挑战性的问题。第四，如何从服务用户的角度来提高用户友善性。软件质量优劣的最终评判者是用户。对用户不友善的软件，用户当然不会乐于使用。为使软件对用户友善，人与系统的接口界面应尽可能采用自然语言与图形语言，这一点颇为重要，采用自然语言与图形语言还只是接口界面的表示问题，更重要的还是接口界面的结构。结构宜简明，力戒烦琐。最后随着非经典计算如量子计算等的研究进展，亦需研究如何为之开发软件。

展望未来，挑战与机遇并存；随着我们对软件开发过程本质认识的深入，软件开发与运行的功效越来越高；软件的智能化水平得到不断的提升，能够不断适应开放、动态与多变环境的需要；软件产品的质量能够从多个方面得到综合化的保障，软件系统的可信性将得到大幅度的提升；软件系统的用户友善性越来越好，能够更好地服务广大用户的需求；总之，软件系统将能够更好地满足信息化社会的需要。

参考文献

1. 徐家福. 软件技术漫谈. 计算机科学，1992，19(1)

2. NCO/NITRD. Future of Software Engineering Research. Report of the 2010 FSE/SDP Workshop on the Future of Software Engineering Research, Santa Fe, New Mexico, USA. 2010

3. Sebastian Nanz (ed.). The future of software engineering. Springer, 2011　（徐家福　吕建）

jisuanji ruanjian de falü baohu

计算机软件的法律保护（legal protection of computer software）　以法律手段对计算机软件的知识产权提供保护和为支持计算机软件的安全运行而提供的法律保护。

计算机软件知识产权是指公民或法人对自己在计算机软件开发过程中创造出来的智力成果所享有的专有权利。包括著作权、专利权、商标权和制止不正当竞争的权利等。

对计算机软件知识产权加以保护是为保护智力成果创造者的合理权益，以维护社会的公正，维护软件开发者成果不应无偿占用的原则，鼓励软件开发者的积极性，推动计算机软件产业及整个社会经济文化的尽快发展。

软件的权利人可拥有以下三方面知识产权：

该软件的表达（例如程序的代码、文档等）方面的权利——著作权；

该软件的技术设计（例如程序的设计方案、处理问题的方法、各项有关的技术信息等）方面的权利——专利权和制止不正当竞争的权利；

该软件的名称标识方面的权利——商标权。

计算机软件的著作权　著作权又称为版权，是指作品作者根据国家著作权法对自己创作的作品的表达所享有的专有权利的总和。1990 年 9 月我国颁布的《著作权法》规定，计算机软件是受著作权法保护的一类作品。1991 年 6 月颁布的《计算机软件保护条例》作为著作权法的配套法规是保护计算机软件著作权的具体实施办法。我国的法律和有关国际公约认为：计算机程序和相关文档、程序的源代码和目标代码都是受著作权法保护的作品。

按照法律规定，软件开发者在一定的期限内对自己软件的表达（例如程序的代码、文档等）享有的专有权利包括发表权、开发者身份权、以复制、展示、发行、修改、翻译、注释等方式使用其软件的使用权、使用许可权和获得报酬权以及转让权。国家依法保护软件开发者的这些专有权利。对软件权利人利益的最主要的威胁是擅自复制程序代码和擅自销售程序代码的复制品，这是侵害软件权利人的著作权的行为。因此，软件的著作权是软件权利人的最主要的权利。

著作权法的原理是保护作品的表达，即作品本身，著作权法不保护作品的构思。对软件的著作权保护不能扩大到开发软件所用的思想、概念、发现、原理、算法、处理过程和运行方法。因此，参照他人程序的技术设计，独立地编写出表达不同的程序的做法并不违反著作权法。不过，对软件进行修改属于软件著作权人的专有权利。如果有人在他人程序著作权有效期内，擅自对他人程序进行修改改编，所产生的程序并没有改变他人程序设计构思的基本表

达，在整体上与他人程序相似，则虽然在代码文字表达方面存在不同，仍属于侵害他人程序著作权的行为。

与计算机软件相关的发明的专利权 专利权是由国家专利主管机关根据国家颁布的专利法授予专利申请者或其权利继受者在一定的期限内实施其发明以及授权他人实施其发明的专有权利。这里所说的发明包括对产品的发明及其改进，也包括对方法的发明及其改进。世界各国用来保护专利权的法律是专利法，专利法所保护的是已经获得了专利权的、可以在生产建设过程中实现的技术方案。各国专利法普遍规定，能够获得专利权的发明应当具备新颖性、创造性和实用性。中国的《专利法》已经于 1984 年 3 月颁布。

一般地说，计算机程序代码本身并不是可以申请发明专利的主题，而是著作权法的保护对象。不过，同设备仪器结合在一起的计算机程序可以作为一项产品发明的组成部分，同整个产品一起申请专利。此外，一项计算机程序无论是否同设备仪器结合在一起，如果在其处理问题的技术设计中具有发明创造，在不少国家里，这些与计算机软件相关的发明创造可以作为方法发明申请专利，很多有关地址定位、虚拟存储、文件管理、信息检索、程序编译、多重窗口、图像处理、数据压缩、多道运行控制、自然语言翻译、程序编写自动化等方面的发明创造已经获得了专利权。在我国，不少有关将汉字输入计算机的发明创造也已经获得了专利权。一旦这种发明创造如果获得了国家专利主管机关授予的专利权，在该专利权有效期内，其他人在开发计算机程序时就不能擅自实施这种发明创造，否则将构成侵害他人专利权的行为。

有关计算机软件中商业秘密的不正当竞争行为的制止权 如果一项软件的技术设计没有获得专利权，而且尚未公开，这种技术设计就是非专利的技术秘密，可以作为软件开发者的商业秘密（Trade secret，也有人译为工商秘密或者营业秘密）而受到保护。把软件的尚未公开的技术设计作为商业秘密保护，是保护软件技术设计的主要途径。一项软件的尚未公开的源程序清单通常被认为是开发者的商业秘密。有关一项软件的尚未公开的设计开发信息，如需求规格、开发计划、整体方案、算法模型、组织结构、处理流程、测试结果等都可被认为是开发者的商业秘密。

对于商业秘密，其拥有者具有使用权和转让权，并可以许可他人使用，也可以将之向社会公开或者去申请专利。不过，对商业秘密的这些权利不是排他性的。任何人都可以对他人的商业秘密进行独立的研究开发，也可以采用反向工程方法或者通过拥有者自己的泄密行为来掌握它，并且在掌握之后使用、转让、许可他人使用、公开这些秘密或者对这些秘密申请专利。然而，根据我国 1993 年 9 月颁布的《反不正当竞争法》，商业秘密的拥有者有权制止他人对自己商业秘密从事不正当竞争行为，这里所称的不正当竞争行为包括：以不正当手段获取他人的商业秘密，使用以不正当手段获取到的他人的商业秘密，接受他人传授或透露了商业秘密的人（例如商业秘密拥有者的职工、合作者或经商业秘密拥有者许可使用的人）违反事前约定，滥用或者泄露这些秘密。

一项信息成为商业秘密的前提在于其本身是秘密。一项商业秘密一旦被公开就不再是商业秘密。为了保护商业秘密，最基本的手段就是依靠保密机制，包括在企业内建立保密制度、同需要接触商业秘密的人员签订保密协议等。

计算机软件名称标识的商标权 对商标的专用权也是软件权利人的一项知识产权。所谓商标是指商品的生产者或者经销者为使自己的商品同其他人的商品相互区别而置于商品表面或者商品包装上的标志，通常用文字、图形或者兼用这两者组成。

国际软件行业现在十分重视商标的使用。有些商标用于标识提供软件产品的企业，如“IBM”，“DEC”，“MS”等，它们是对应企业的信誉的标志。有些商标则用于标识特定的软件产品，如“UNIX”，“OS/2”，“Lotus 1-2-3”等，它们也是特定软件产品的名称，是特定软件产品的功能和性能的标志。

在一般情况下，一个企业的标识或者一项软件的名称未必就是商标。然而，当这种标识或者名称在商标管理机关获准注册、成为商标后，在商标的有效期内，注册者对它享有专用权，他人未经注册者许可不得再使用它作为其他软件的名称。否则，就构成冒用他人商标、欺骗用户的行为。很多国家颁布了商标法以保护商标注册者的这种专用权，我国的《商标法》已经在 1982 年 8 月颁布。 （应明）

jisuanji shijue

计算机视觉（computer vision） 研究使机器具有“看”的能力的一门技术。狭义地讲，计算机视觉就是通过分析摄像机获取的图像来达到对物体形

成表达的科学和技术，与图像理解，视频分析，**模式识别**有密切的联系，但都有所区别。

粗略地讲，计算机视觉经历了以下 4 个主要发展历程：

1. 马尔视觉计算理论　1982 年马尔视觉计算理论（参见**视觉计算理论**）的提出，标志着计算机视觉成为了一门独立的学科。马尔视觉计算理论包含两个主要观点：首先，马尔认为人类视觉的主要功能是复原三维场景的可见几何表面，即三维重建问题；其次，马尔认为这种从二维图像到三维几何结构的复原过程是可以通过计算完成的，并提出了从图像初始略图（sketch）→物体 2.5 维描述→物体 3 维描述一套完整的计算理论和方法。

2. 主动视觉，目的视觉　针对马尔理论在具体应用中遇到的困难，以 Y. Aloimonos，R. Bajcsy 等为代表的学者在 20 世纪 80 年代末、90 年代初提出了“目的视觉”，“主动视觉”，“定性视觉”等理论。这些理论的共同特点是认为，马尔理论从下到上的三维重建过程由于缺乏目的性，缺乏高层知识反馈，从而导致三维重建框架不可行，重建算法不鲁棒。他们强调视觉算法高层知识反馈的必要性和重要性，以及视觉主体与环境交互的重要性。这些理论尽管从原理上来说更加符合人类视觉过程，但由于“利用什么样的高层知识”，“如何利用高层知识”，“视觉主体如何与环境进行有效交互”等这些核心问题目前人们还知之甚少，在近期内估计还很难建立有效的计算模型，所以这些理论自从 1994 年在 CVGIP：Image Understanding 上组织关于主动视觉的大辩论后，至今仍没有显著性进展。

3. 分层重建理论（stratified reconstruction）　以 O. Faugeras 和 R. Hartley 等提出的分层重建理论是 20 世纪 90 年代计算机视觉领域最活跃的研究内容。分层重建的基本思想是指从图像到三维欧几里得空间的重建过程中，先从图像空间得到射影空间下的重建（11 个未知数），然后将射影空间下重建的结果提升到仿射空间（3 个未知数），最后将仿射空间下重建的结果提升到欧几里得空间（5 个未知数）。这种分层重建方法由于每一步重建过程中涉及的未知变量少，几何意义明确，所以算法的鲁棒性得到了有效提高。

4. 基于学习的视觉　物体识别是计算机视觉的重要研究内容。随着基于图像的物体表达（View Based Representation）的提出和**机器学习**的进展，基于学习的视觉是近年来计算机视觉的研究热点。所谓基于学习的视觉，就是指利用机器学习的手段来对图像物体进行识别的方法，包括对物体的识别和分类（object identification and categorization）。

另外，视觉伺服，大场景三维重建，图像标注，特定领域视频分析，视觉监控都是近年来计算机视觉领域非常活跃的研究方向。

参考文献

1. Marr D，Vision：A computational Investigation into the human representation and processing of visual information. W. H. Freeman and Company，1982

2. Aloimonos Y. Active vision revisited. In：Y. Aloimonos（ed.）Active Perception，Lawrence Erlbaum Associates，1993

3. Hartley R，Zisserman A. Multiple view geometry in computer vision. Cambridge University Press，2000

4. Poggio T，Bizzi E. Generalization in vision and motor control. Nature，2004，431（14）：768-774

（胡占义）

jisuanji shijue zhong de jiegouguang fangfa

计算机视觉中的结构光方法（structured light method in computer vision）　用结构光和摄像机来测量场景三维信息的方法。结构光测量系统与**立体视觉**系统类似，只是将其中的一个摄像机换成产生结构光的投影仪，如图 1 所示。常用的投影仪有 LCD 投影仪、激光扫描仪等。

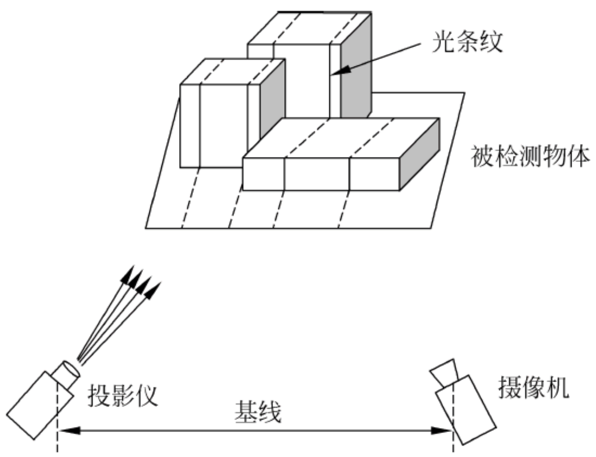


图 1　结构光测量系统

结构光（通常指一组有规律的光点或光线）方法的基本思想是投影仪产生特定模式的几何图案投射到场景上，摄像机（可以采用多个摄像机）采集场

景几何图案图像,根据三角测量原理计算几何图案上每一点与摄像机之间的距离,得到场景稠密深度图(dense depth map)。与立体视觉方法相比,结构光方法避免了复杂的对应性求解问题,也无须考虑诸如物体几何形状、反射特性、纹理、照明等因素,具有计算速度快、精度高、鲁棒性好、成本低等特点,广泛应用于实际场景三维测量中。

投影仪产生的结构光几何图案有斑点、线条、编码等。斑点结构光方法利用光点扫描整个场景,得到每个点的深度。该方法鲁棒性好、准确度高,但测量速度慢。线结构光方法提高了测量速度,却增大了对应关系的确定难度,影响了测量准确度;编码结构光法采用投影仪向场景投射一幅或多幅经过编码的图案,然后采集图案图像并进行解码来测量三维场景,这种方法测量速度快,识别率高,是目前结构光法发展的主流方向。编码结构光法可以分为时分编码、空间邻域编码和直接编码三种方法。时分编码将不同的编码图案按时序先后(如由粗到细)投射到被检测物体表面,按序组合所有编码图案进行编码。该方法具有很高的精确度和分辨率,但由于按时序进行编码,难以用于动态测量。空间邻域编码将所有编码图案聚集成一种唯一的图案,图案中的点通过其自身与邻域的信息共同表示,故它可以直接对照编码图案与编码方式进行解码,可以用于动态测量。该方法主要依赖于对邻域的识别,会受到物体表面特性影响(如反射率、颜色等),加大了解码难度。直接编码是直接根据每个编码像素的灰度或者颜色来识别,方法简单,理论上可以获取高分辨率的三维信息。实际应用中,直接编码对噪声很敏感,测量准确度较低。

目前已经出现了不少结构光三维测量实时系统,测量速度达到视频速度(30 帧/秒),有的系统达到 100 帧/秒,完全满足实际中动态场景测量的要求,比如动态三维手势、三维人脸及人脸表情的测量。

参考文献

Salvi J, Pages J, Batlle J. Pattern codification strategies in structured light systems. Pattern Recognition, 2004, 37: 827-849

(贾云得)

jisuanji shuxue

计算机数学(computer mathematics) 研究用计算机进行数学问题符号演算和推理的学科。计算机数学是随着计算机科学和数学的发展而逐步融合形成的。由于计算机科学和计算机的出现与发展,使得数学问题的机械计算和证明成为实际可能,并且反过来大大推动了计算机科学和数学的发展。

计算机数学和作为数学一个分支的计算数学既有区别又有联系。历史上,计算数学是以数学分析和**代数学**为基础发展起来的近似计算方法,与计算机数学有着不同的起源背景,它的输入和输出都是数值,在研究问题的目的、方法和技巧方面都形成了自己的体系。而计算机数学的输入和输出都是数学表达式,所涉及的是符号演算。例如计算方程

$$x^3 + 3x^2 + 3x + 2 = 0$$

的解,计算机数学的方法给出精确解(解析解)

$$x_1 = -2, \quad x_2 = \frac{-1 + \sqrt{-3}}{2}, \quad x_3 = \frac{-1 - \sqrt{-3}}{2},$$

而数值计算的方法给出近似解(数值解)

$$x_1 = -2, \quad x_2 = -0.5 - 0.866025\mathrm{i},$$
$$x_3 = -0.5 + 0.866025\mathrm{i}$$

数学中的很多问题都属于非数值计算,例如计算不定积分,计算一个理想的准素分解,推导数学定理等,在这个领域,计算机数学有很广阔的理论和应用发展前景,即使在传统的数值计算工作中,有时也需要借助计算机数学的手段,以丰富和改善数值计算的结果。计算机数学的内容涵盖了数学的许多方面,例如**定理机器证明(自动推理)**、**计算机代数**、**计算几何**等,并伴随日益增多的应用。使用计算机数学辅助设计机械结构,不仅可以完整地描述机构的运动状态,而且可以发现在一些极端条件下隐含的缺陷,大大弥补了传统设计方法的不足。使用计算机数学开发的机器人运动控制程序,可以从运动学和动力学的角度设计出最佳的运动路线,避开障碍物。

用计算机解决数学问题,关键是设计相关的算法,从这一点上说,计算机数学就是研究各种数学问题的算法。尽管计算机本质上只能处理有限的对象,在一定意义上,计算机数学也能驾驭无穷对象,处理涉及到连续对象的一些数学问题。

数学的进步总是伴随着计算方法的完善和证明能力的提高。但只是在计算机问世以后,才真正刺激了数学问题算法设计的快速发展。1958 年,王浩设计了一个计算机程序,在 9 min 的时间里,证明了罗素和怀德海《数学原理》中几百条定理。并提出了"数学机械化"的概念,这是计算机数学作为一个

独立研究领域的开端。从此以后，计算机数学在数学的各个领域迅速渗透，随着计算机科学和技术的发展，一些阻碍计算机数学应用的问题被逐渐克服，例如运算的速度、存储器的容量、软件开发的周期等，使得利用计算机辅助数学研究成为常规的手段。例如在有限单群分类的研究中，对于散在单群的寻找，计算机就起到了关键的作用。

由于代数学侧重对数学对象结构上的刻画，以及代数学所具有的有限性、算法性和构造性等特点，因此在设计数学问题算法时，经常把问题转化为代数形式，计算机代数自然成为计算机数学的核心内容。例如在初等几何定理证明中，就是通过坐标的引进，将初等几何问题翻译成多项式方程组的形式，从而把定理的成立与否转换为对于多项式方程组零点结构的判定问题。又例如在微分方程解析解的计算中，也是将微分方程解的结构问题转化为微分代数或多项式代数的语言描述形式来设计相应的算法。

计算机代数的主要内容有归约计算、多项式计算和计算微分代数等几大部分，这里简单介绍域上多项式方程组（一元的和多元的）零点的计算和结构表示问题。

对于单变元多项式方程，四次（含四次）以下的可以根据求根公式写出符号解。四次以上的多项式没有一般的求根公式，目前计算机代数软件中实用的方法是将$f(x)$通过因式分解表示为两个次数较低的多项式的乘积$f(x)=g(x)h(x)$，再分别求出每一个因式的根；或者使用复合式分解，将$f(x)$写成两个次数较低的多项式的复合$f(x)=g(h(x))$，然后通过计算$g(x)$和$h(x)$的根得到$f(x)$的根。通过辅以其他一些技巧，这个方法在很多情况下是有效和快速的。对于不存在公式解的方程，使用该方法自然是失效的。但在有些情况下，即便公式解存在，该方法也可能失败，因此该方法不是完备的。

多元多项式方程组的零点集一般是代数曲面，不是孤立的点，因此零点集的表示和计算都是复杂的问题，除了在一些特定情况下的算法外，在一般意义下给出零点计算和结构表示的方法首推吴方法和Groebner基方法。

吴方法：设

$$F=\{f_1(x_1,x_2,\cdots,x_n),f_2(x_1,x_2,\cdots,x_n),\cdots,f_m(x_1,x_2,\cdots,x_n)\}$$

是域A上的多元多项式的集合。多项式$f(x_1,x_2,\cdots,x_k)$中所包含变量的最大下标k称为$f(x_1,x_2,\cdots,x_k)$的类，记做Class(f)。从F中取出一个尽可能大的子集

$$G=\{g_1,g_2,\cdots,g_p\}$$

使得$\mathrm{Class}(g_1)<\mathrm{Class}(g_2)<\cdots<\mathrm{Class}(g_p)$，这个子集称为$F$的基列，用伪除法对$G$去逐个除$F$中的$f$，对于每一个$f$，得到表达式

$$Bf=h_1g_1+h_2g_2+\cdots+h_pg_p+R$$

其中B是g_i的初式（首项系数）$\mathrm{In}(g_i)$的幂的乘积，$B=\mathrm{In}(g_1)^{\rho_1}\mathrm{In}(g_2)^{\rho_2}\cdots\mathrm{In}(g_p)^{\rho_p}$，$R$称为伪余式。如果$R\neq0$，将$R$添加到$F$中得到$F'=F\cup\{R\}$，如果对于某个$g_k$，$\mathrm{Class}(R)=\mathrm{Class}(g_k)$，则用$R$替换$g_k$，如果对所有的$g_k,1\leqslant k\leqslant p$，$\mathrm{Class}(R)\neq\mathrm{Class}(g_k)$，则将$R$添加到$G$中去，得到的新的集合仍记做$G$，继续用新的$G$去除$F'$中的每一个多项式。该过程必终止（所有的$R=0$），终止时的多项式集合记做$\bar{F}$，基列记做$\bar{G}=\{\bar{g}_1,\bar{g}_2,\cdots,\bar{g}_q\}$，对于$\bar{F}$中的任何多项式$f$，满足

$$\bar{B}f=h_1\bar{g}_1+h_2\bar{g}_2+\cdots+h_q\bar{g}_q,$$

这样的基列称为$\bar{F}$的特征列，$\bar{F}$与F生成的理想是相同的。记$Z(F)$为多项式集F的零点集，则

$$Z(\bar{G})\setminus\left(\bigcup_{1\leqslant i\leqslant q}Z(\mathrm{In}(\bar{g}_i))\right)\subseteq Z(\bar{F})=Z(F)\subseteq Z(\bar{G})$$

特征列中的多项式可以按含变量的个数排成三角形式

$$\bar{G}=\begin{Bmatrix}f_0(x_1,x_2,\cdots,x_d)\\ f_1(x_1,x_2,\cdots,x_d,x_{d+1})\\ \vdots\\ f_{n-d}(x_1,x_2,\cdots,x_d,\cdots,x_n)\end{Bmatrix}$$

零点的结构和性质容易从$\bar{G}$导出，在零点个数有限的情况下（$d=1$），零点还可以通过逐个方程的计算迭代求出。吴方法把复杂的方程组化成了简单明了的表示形式，并保留了方程组零点的许多重要性质，适应于以零点论式表述的数学问题，在几何问题计算、机器定理证明等领域能够借助吴方法设计快速和有效的算法。

Groebner基方法：设

$$F=\{f_1(x_1,x_2,\cdots,x_n),f_2(x_1,x_2,\cdots,x_n),\cdots,f_m(x_1,x_2,\cdots,x_n)\}$$

是域A上的多项式的集合。对于F中多项式$f(x_1,x_2,\cdots,x_n)$，记

$$f(x_1,x_2,\cdots,x_n)=\mathrm{head}(f)+\mathrm{tail}(f),$$

其中head(f)是f的首项，$\mathrm{tail}(f)=f-\mathrm{head}(f)$。取$F$中任两个多项式$f,g$，构造

$$h=\frac{\mathrm{lcm}(\mathrm{head}(f),\mathrm{head}(g))}{\mathrm{head}(f)}f-$$

$$\frac{\mathrm{lcm}(\mathrm{head}(f),\mathrm{head}(g))}{\mathrm{head}(g)}g$$

lcm 表示最小公倍式，如果存在多项式 $p \in F$，使得 head(h)是 head(p)的倍式，即 qhead(p) = head(h)(q 是单项式)，则令 $h' = h - qp$(这一步称为首项归约)，如此继续，直到取得多项式 $h_{f,g}$，$h_{f,g} = 0$ 或者 head($h_{f,g}$)不是 F 中任何多项式首项的倍式。$h_{f,g}$ 称为 f 和 g 的 s-多项式，若 $h_{f,g} \neq 0$，将 $h_{f,g}$ 添加到 F 中得到新的 F'，继续对 F' 中的所有多项式对(f,g)计算 s-多项式 $h_{f,g}$，并将其添加到 F' 中，这个过程最终必停止，停止时得到的多项式集合记做 G，它是有限的，称为 F 的 Groebner 基。记

$$\mathrm{Head}(F) = (\{\mathrm{head}(f) \mid f \in F\})$$

即由 F 中的多项式的首项生成的理想(称为 F 的首项理想)，G 满足两条特征性质：①$(G) = (F)$；②$\mathrm{Head}(F) = \mathrm{Head}(G)$。

一般地，G 中包含的多项式是很多的，但 F 与 G 的零点集是完全相同的，$Z(F) = Z(G)$，G 也可以表示为三角形式

$$G = \begin{Bmatrix} F_0(x_1,\cdots,x_d) \\ F_1(x_1,\cdots,x_d,x_{d+1}) \\ \vdots \\ F_{n-d}(x_1,\cdots,x_d,\cdots,x_n) \end{Bmatrix}$$

但与吴方法不同的是，这时 $F_i(x_1,\cdots,x_d,\cdots,x_{d+i})$ 是多项式集合。Groebner 基也给出了原方程组零点结构和性质的很好刻画。在零点个数有限的情况下($d=1$)，零点也可以通过逐步迭代的方法求出。由于 Groebner 基 G 保持了原多项式集合 F 的理想性质，适应于以理想论式表述的数学问题，例如代数学中关于理想的各种计算。

多元多项式方程组零点的计算是计算机数学中最基本的问题之一，很多理论和实际问题都可以转化为多元多项式方程组零点计算与表示，或者相关的理想结构的计算，因此在很多应用场合都能够遇到多元多项式方程组的计算问题，寻求更加快速高效的算法仍是当前研究的重点。

计算机数学的产生和发展，使得数学演算和推理更加机械化和形象化，催化了数学与其他科学和工程技术的结合，传统数学的很多抽象研究成果，借助计算机这种有力的工具，找到了新的应用手段，从而大大推动了计算机科学与数学自身的发展，以及在其他领域的更为广泛的和富有成效的应有。

参考文献

1. 林东贷，李文林，虞言林. 数学与数学机械化. 济南：山东教育出版社，2001

2. 吴文俊. 王者之路——机器证明及其应用. 长沙：湖南科学技术出版社，1999

3. Blum L, Cucker F, Shub M, Smale S. Complexity and real computation. New York: Springer-Verlag, 1998

(李廉)

jisuanji suanshu luoji yunsuan

计算机算术逻辑运算(computer arithmetic/logic operation) 计算机对数据所进行的算术运算和逻辑运算。

随着计算机应用范围的不断扩大，计算机能表达、传送、存储和处理的数据有数字、字母、符号、图形、图像和声音等多种形式。尽管数据的表示形式、处理方法以及解决问题的算法有很大差别，但最终可以分解为一组或若干组遵循一定次序关系的算术运算和逻辑运算。因此算术运算、逻辑运算以及控制这些运算的机制构成了计算机运行的基础。

人们习惯于用十进制记数法来表示数的数值，但在计算机内部运算时通常采用二进制记数法，这是因为二进制数的每一位只有“1”或“0”两个值，它们可用电子线路(如触发器)的两种状态来表示。二进制运算规则比较简单，也便于在计算机中实现。一般人机交互的数据是以十进制形式表示的，而在计算机内部则经过转换后变为二进制形式。有的计算机可直接对十进制数进行运算(参见**十进制算术运算**)。

计算机的基本算术运算有加法、减法、乘法和除法运算，为了便于实现加减法运算，除了一般的表示方法(原码)以外，又采用了补码和反码来表示数(参见**数制**)。为了使数有不同的精度和范围，又有定点数和浮点数两种表示方式，并制定了浮点数标准。对二进制数如何进行加、减、乘、除运算可参见**二进制算术运算**。

一个二进制码只有“1”和“0”两个值，如果赋以逻辑属性，例如表示成“真”和“假”，或者表示成“是”和“非”，将这种二进制码表示的变量称为逻辑变量，描述逻辑变量关系的函数称为逻辑函数，对逻辑变量和逻辑函数进行描述和运算的数学工具称为逻辑代数，也称为**布尔代数**，实现逻辑功能的电路称为逻辑电路。逻辑电路除了用于计算机的算术逻辑运算部件中以外，在计算机的其他部件中也广泛应用。有关基本逻辑运算的规则及逻辑电路的表示方法可参见**逻辑运算**。

(王爱英)

jisuanji tixi jiegou

计算机体系结构(computer system architecture) 计算机的外在组织结构与功能行为。外在组织结构指的是计算机软件人员所能理解的计算机的一级组成部分所构成的集合及其各元素之间的联系。功能行为指的是计算机软件人员所能理解的计算机的基本功能。这些基本功能集中体现在计算机指令系统之中。指令系统既是计算机硬件功能的集中体现,又是计算机软件人员赖以构建软件的基础。

计算机体系结构研究者的任务是设计一台计算机,满足功能需求以及价格、能耗、性能、可用性等目标。从研究内容来看,计算机体系结构包括计算机设计中的两个方面:指令集体系结构和微体系结构。

作为一门学科,指的是以上述实体含义的计算机体系结构为研究对象的学科,或者说,在实体含义的计算机体系结构的研究与开发中所涉及的所有理论、原则、方法、技术所构成的学科。

发展简史

“计算机体系结构”一词来源于英文 computer architecture,也有人译成“计算机系统结构”。Architecture 原来用于建筑领域,其意义是“建筑学”“建筑物的设计或式样”,它是指一个系统的外貌。“计算机体系结构”一词已经得到了普遍应用。

粗略地讲,计算机体系结构以及相关的软硬件开发模式经历了如下四个阶段。

第一阶段是 20 世纪 40 年代中到 50 年代末。在计算机体系结构上面,已经出现了两种不同的结构,一种是哈佛结构,另一种是冯·诺依曼结构。哈佛结构将数据和指令分别存于不同的存储器中,而冯·诺依曼结构则使用统一的存储介质来存放。后者对于现代的处理器设计具有较大影响,现代处理器结合了这两个计算机体系结构的特点,当然主要还是采用了冯·诺依曼结构。

第二个阶段是从 20 世纪 50 年代末到 60 年代中期。在计算机体系结构方面,开始采用补码的方式进行整数的表示和运算,并且可以进行浮点数的运算。在计算速度上,将指令的执行时间从毫秒级降到微秒级。在软件上开始出现了高级语言,并开始出现了操作系统,能够每次处理一个程序。IBM 的 7000 系列大型机是这个时期的代表,完全使用晶体管构造。

第三个阶段是从二十世纪 60 年代中期到 70 年代中期。在计算机体系结构上,出现了缓存内存形式以提高整个计算机的运行性能。在上层的系统软件以及应用程序上出现了分时共享系统,计算机图形以及结构化编程方法等的软件结构。这个时期的计算机的典型代表就是 IBM 的 360 系统,该系列计算机第一次将计算机体系结构以及相应的实现进行了区分,有了兼容性的概念。在相同的计算机体系结构条件下,也可以具有不同的实现以获得更高的速度,但是在软件上还是能够兼容原先的软件,维护现有软件的投资。

第四个阶段则是当代计算机体系结构大发展的时期,从二十世纪 70 年代中期至今。随着大规模商用计算机的大量涌现,出现了非常重要的个人计算机。通过增加电路板上晶体管的密度,性能、性价比、功耗等各个方面都不断发展。计算机网络能够将不同地理位置的计算机连接在一起,构成一个超大规模的计算系统。流水线、多发射超标量、向量处理等各个计算机体系结构上的技术在这个时期得到充分发展,将计算机处理器的性能推到了一个前所未有的高度。同时,互联网络的发展也将计算机的应用深入到整个人类社会的方方面面。特别是,随着互联网以及计算机的紧密结合与发展,计算机体系结构已迅速地从集中的主机环境转变成分布的客户机/服务器(或浏览器/服务器)环境,网格计算环境、对等计算环境等在地理位置上具有广泛分布的计算环境。世界范围的信息网为人们进行广泛交流和资源的充分共享提供了条件。新兴的云计算技术为分布式计算和服务提供形式带来了新的气象,计算中心的服务开发与部署很容易惠及大众。

计算机体系结构包含的内容

按照现代观点,计算机体系结构包含指令集体系结构和微体系结构两部分内容:

指令集体系结构(ISA:Instruction Set Architecture)是计算机系统的硬件与软件之间的接口,无论是系统软件的编程人员还是应用软件的编程人员,都需要根据指令集体系结构来编写**软件系统**。指令集系统结构包括的内容不仅仅是指令的相关内容,还包括诸如**寻址方式**,寄存器等。如果已有大量软件是针对某个指令集体系结构设计,架构师可能会选择让新设计的计算机实现已有的指令集。

微体系结构(microarchitecture,或称为“组成”)解决如何通过底层的数据通路、数据处理单元以及寄存器暂存部件等相关硬件组成部分来实现某种指令集体系结构。其内容涵盖**主存储器**系统、主存互

连以及内部处理器的设计等计算机设计的高级方面。微体系结构所包含的内容要比指令集体系结构更加丰富。一个明显例子是处理器上的**高速缓冲存储器**(cache),它在指令集体系结构中并不涉及,但在微体系结构中却占有非常重要的地位。有着相同指令集体系结构的两部处理器可以有不同的微体系结构。典型例子是 AMD Opteron 和 Intel Core i7,两者都是实现的 x86 指令集,但是它们有非常不同的流水线和高速缓冲存储器(cache)组织。

计算机体系结构发展趋势

近年来,计算机体系结构发展迅速。第一个趋势是多核以及众核处理器不断发展。随着处理器频率发展的瓶颈以及晶体管密度发展瓶颈的到来,多核及众核处理器成为处理器体系结构的研究重点。多核和众核处理器也会集成越来越多的功能,如集成虚拟化的特性,集成内存访问单元等。另一个发展趋势是对数据密集型体系结构的探索。随着数据密集型计算应用需求的普及,数据密集型体系结构越来越受重视。一个例子是采用在存储设备上增加计算单元来提高数据处理速度。第三个发展趋势体现在多机系统体系结构的发展,出现了新形式的软硬件体系结构。当前的一个大的发展趋势是对云计算等新型的计算模式的研究与开发。

参考文献

1. 郑纬民,汤志忠. 计算机体系结构. 北京: 清华大学出版社,1998

2. Hennessy J L, Patterson D A. Computer architecture: a quantitative approach. 5th ed. Burlinton, MA: Morgan Kaufmann Publishers, 2011

(郑纬民)

jisuanji tuxing biaozhun

计算机图形标准(computer graphics standard)

计算机系统中应用程序与图形软件、图形软件与图形硬件之间的接口标准,以及用于记录图形信息的数据文件格式的标准。

计算机图形技术在**计算机辅助设计**、办公自动化、**地理信息系统**、过程控制、仿真、图像处理、软件开发等领域中起着重要的作用,它已经成为人机交互的一种主要手段。但由于图形输入、输出设备种类多,工作原理和性能参数差别大,因此图形应用程序的开发难度大、成本高,比较难以具备良好的易移植性。

图形应用程序的易移植性意味着4个方面的要求:①应用程序在不同系统之间的易移植性;②应用程序与图形设备的无关性;③图形数据的易移植性(互操作性);④程序员的易移植性,即程序员不经重新培训就能为不同系统编制图形应用程序。为了实现这些要求,系统中有3个接口必须实现标准化。

首先是应用程序与图形软件的接口,这是一个应用程序接口(API)。图形软件是一组有关图形处理的常用子程序的集合,它有效地隔离了应用程序与图形硬件设备之间的联系。因此,该接口的标准化就能实现应用程序在源程序级的易移植性。20世纪90年代,确立了3个国际标准,分别是图形核心系统 GKS(ISO/IEC 7942: 1994)、三维图形核心系统 GKS-3D(ISO 8805: 1988)以及程序员的层次式交互图形标准 PHIGS(ISO/IEC 9592: 1997)。这3个标准中 GKS 是二维图形标准,GKS-3D 和 PHIGS 是三维图形标准,它们既有许多共同之处,又各有特色,互相补充。国际标准化组织还对上述3个标准分别制定了与 FORTRAN、Pascal、C 和 Ada 语言的语言联编标准。语言联编是图形功能调用的子程序名及参数的约定,它为应用程序开发人员提供了共同遵循的准则,目的同样是为了应用程序具有良好的易移植性。

由于国际标准制定的迟缓以及符合标准的图形软件速度慢、内存开销大,以上三个标准逐渐失去了生命力,淡出了工业应用范畴。在另一方面,一些跨国图形软硬件公司因自行开发了图形软件 API,并在应用中得到了普遍认可,成为事实上的工业标准。例如,在工业设计与游戏领域 Silicon Graphics Inc (SGI)公司的 OpenGL 标准,以及 Microsoft 公司所推出的 Direct3D 系列 API 等。这两个事实标准与 GKS 标准有很大的相关性,都具有多种语言实现能力,而 OpenGL 具有更强的移植性。由于 OpenGL 的广泛使用,各大公司联合起来组成了 Khronos 国际工作组,负责该标准的维护和新版本发布。

其次是图形软件与图形硬件之间的接口,它也是一个程序接口,称为图形工作站接口。实现这个接口的标准化,就可以保证图形软件与图形设备之间的相互独立性。ISO 为这个接口制定的标准称为软件与图形设备对话的接口技术,即 CGI(ISO/IEC 9636: 1991)。CGI 也可以作为不同图形设备之间的数据交换标准,它规定了 CGI 格式的数据流标准(ISO/IEC 9637: 1994),符合标准的数据可以在具有 CGI 接口的设备(如图形显示器、绘图机等)上直接进行显示或打印。

Microsoft 公司在 Windows 操作系统中提供的图形设备接口 GDI 以及跨平台图形开源项目 SDL,均可视为图形软件与图形硬件之间的接口,其功能是在显示器、打印机等图形设备上提供对文本、图形和图像输出的支持。这些接口屏蔽了不同输出设备各自的物理特性,使得应用程序能够在它所支持的任何图形输出设备上运行。

其三是数据接口,它规定了记录图形信息的数据文件格式的标准。实现了图形文件格式的标准化之后,程序与程序或系统与系统相互交换图形数据就有了保证。ISO/IEC 制定的用于图形数据交换的国际标准是 CGM(ISO/IEC 8632：1999)。CGM 定义了二维图形(图片)信息的计算机可解释的表示,这种表示与任何特定设备或应用无关,其目的是便于图形数据的储存、检索以及在不同系统、不同应用和不同设备之间的交换。近些年来,CGM Open 联盟与 W3C 合作开发了 CGM 的一种框架——WebCGM,使 CGM 图形元文件更适合在 Web 环境下应用。与 WebCGM 相比,由 W3C 制定的 SVG 标准使用 XML 语言描述二维矢量图形,在二维图形数据文件领域获得了更为广泛的应用,并且获得诸多浏览器以及移动硬件的支持。

实用中的图形数据交换格式还有多种。例如,在 CAD/CAM 软件中用于交换产品定义数据的美国国家标准 IGES(现已停止发展);法国和欧洲采用的工程数据交换标准 SET(现已被 STEP 取代);ISO 制定的产品数据表示与交换的国际标准 STEP(ISO 10303)中的几何与拓扑信息的表示(ISO 10303—42：2000);ISO 制定的在办公文档中嵌入矢量图形的标准(ISO 8613—8：1994)GGCA。一些厂商也为自己的图形软件制定了专用的数据交换格式,如 Autodesk 公司的 DXF 格式、Microsoft 公司的 WMF 格式等,它们也得到了广泛的使用。

由于虚拟现实以及游戏娱乐等应用领域的迅猛发展,制定了 VRML/X3D 系列标准。VRML 开始是作为一种专为 WWW(**万维网**)而设计的三维图像置标语言。VRML 全称是虚拟现实建模语言,起源于 SGI 工作站上的应用软件 Inventor,是由 VRML 协会(后更名为 Web3D 联盟)设计并制订标准的。VRML 标准中既定义了描述三维模型的编码格式,也定义了描述交互或脚本的编码及行为模式。通过引入 XML 语言描述,VRML 标准进化升级为了 X3D 标准。目前 X3D 标准已通过 ISO 认证。

为了使各个图形标准规范化,明确各个图形标准之间的关系,ISO/IEC 还制定了一个 CGRM 标准(ISO/IEC 11072：1992),它并不针对图形系统中某一特定部分,而是整个图形系统的一个 5 层概念模型,目的是对复杂的系统从整体方面进行约束,使得各个单元技术的发展能够协调一致,可以说,它是一个图形标准的标准。

计算机图形标准已被工业部门和计算机用户所接受,它们已得到广泛的应用,已制定的图形标准将逐步完善,新的图形标准还将继续出现。

参考文献

ISO/IEC 11072：1992. Information Technology-Computer graphics-Computer Graphics Reference Model, 1992

（张福炎　张宏鑫）

jisuanji tuxingxue

计算机图形学(computer graphics)　应用计算机生成、处理和显示图形的理论、方法和技术。它可以生成现实世界中已有景物的图形,也可以生成虚构场景的画面。近年来,由于**多媒体技术**、影视特技、可视分析、**增强现实**等的迅速发展,计算机图形学和**数字图像处理**、**计算机视觉**的结合日益紧密,且相互渗透,进一步促进了相关领域的发展。

早在 20 世纪 50 年代末至 60 年代初期,就出现了图形显示器。1963 年,美国麻省理工学院的 I. E. Sutherland 在他的博士论文中首次提出了交互式计算机图形学的概念。在他开发的 SKETCHPAD 系统中,不仅引入了分层存储符号和图素的数据结构,还可用光笔实现选择、定位等交互功能。50 年来,计算机图形学有了飞速的发展。在图形显示设备方面由 60 年代中期的随机扫描刷新式矢量显示器,70 年代的存储管式显示器,到 80 年代后广泛采用的光栅扫描显示器,价格越来越低廉,显示功能越来越强大,性能越来越稳定。再加上个人计算机性能迅速提高和出现了低价位普及型的**计算机辅助设计**(CAD)及图形生成软件,使得计算机图形广泛融入到人们的科研、生产、工作、学习和日常生活中,成为加强信息传递和理解的有力工具。

计算机图形学目前的主要应用领域有：①计算机辅助设计和制造；②科学计算可视化；③**虚拟现实**与仿真；④计算机动画；⑤计算机艺术；⑥**计算机游戏**；⑦办公自动化及电子出版；⑧**地理信息系统**；⑨影视特技与广告制作等。

计算机图形学的主要内容有**造型技术**、图形绘制及**人机交互技术** 3 部分。要在计算机屏幕上生成

三维景物的图像,首先必须在计算机中建立该景物的表示。这一技术称为造型技术。最常用的**几何造型**技术通过建立描述景物几何形状和拓扑结构的数据结构来表示所要显示的形体。几何形状的可控性及覆盖面是研究工作的重点。近年来,随着三维扫描技术的发展,一些外形复杂的景物大都改由大量三角形面片组成的多面体来表示。由于景物中三角形数量十分庞大,影响了画面实时绘制的速度和景物模型在网络中传输的效率,因此,基于三角形网格的景物**多分辨率造型**技术和几何数据压缩、传输技术成了几何造型近年来研究的热点。**基于图像的造型**技术则通过从一系列不同视点、视线方面拍摄的场景图像(含深度图像)来表示一个场景,进而生成场景在任意新视点处的画面。动画技术的发展,导致了**基于物理的造型**技术。自然界的许多景物很难或者根本无法用规则形体进行表达,如山脉、杂草、毛皮、云彩、波浪、浓雾等。为了表示这类自然景物,又出现了分数维造型、基于文法的造型等。

建立了景物的表示后,还需要根据观察者当前的视点位置和视线方向,生成该景物的真实感画面,这一过程称为**绘制**,它包括景物的取景变换、视域裁剪、消除当前画面中不可见的景物隐藏面,以及基于一定**光照模型**的景物表面光亮度计算等。光照模型包括局部光照模型和整体光照模型,它们都是根据物理和光学的定律对景物表面光照明效果及表面对入射光的吸收、反射、折射情况的描述。常用的真实感图形绘制算法有扫描线算法、光线跟踪算法、光能辐射度方法等。一般来说,生成的景物画面逼真度越高,所需的计算时间就越长,因此,如何实现复杂景物真实感图形的实时动态显示就成为计算机图形学领域多年来追求的目标和研究的关注点。除了提高图形算法的效率以外,运用可编程的图形处理单元(GPU)实现硬件加速是当前普遍采用的方式。

从本质上说,图形绘制是为了增强对场景信息的传递,促进人们对画面中所关注细节的了解。近年来,一种新的绘制技术——表意式绘制(expressive rendering)被提了出来。表意式绘制并不追求真实的光照效果和投影效果,而是讲求核心信息表达和传递的有效性。

三维景物造型、场景绘制都需要在一个操作方便、易学易用的用户界面下工作,包括:图元及造型方法的交互选择;形体、模型的交互操作;观察点、观察方向的交互设置;光照模型参数的交互选取;色彩的交互设定等。这就需要依据人机交互技术的理论和方法,设计出友善的用户界面以满足交互地生成图形的需要。近年来迅速发展的虚拟现实技术,是造型技术、真实感图形实时生成技术及和谐人机交互等多种技术的综合与集成。

参考文献

1. 唐荣锡,汪嘉业,彭群生,汪国昭,等. 计算机图形学教程. 修订版. 北京:科学出版社,2000

2. 孙家广,等. 计算机图形学. 3 版. 北京:清华大学出版社,1998

3. Rogers D F. 计算机图形学的算法基础(原书第 2 版). 石教英,彭群生,等译. 北京:机械工业出版社,2002

(彭群生)

jisuanji wangluo

计算机网络(computer network) 地理上分散的多台互连自主计算机的集合。计算机互联必须遵循约定的通信(网络)协议,由通信设备、通信链路及网络软件实现。计算机网络可实现信息交互、资源共享、协同工作及在线处理等功能。

发展简史

计算机的应用普及是从**微型计算机**的出现开始的。而进一步的发展阶段乃是网络,人们称网络就是计算机,深刻地反映了网络在计算机发展史中极为重要的作用和影响。

1969 年美国国防部的国防高级研究计划署(DARPA)建立了全世界第一个分组交换网 ARPANET,即 **Internet** 的前身。这是一个只有四个节点的存储转发方式的分组交换广域网,是为了验证远程分组交换网的可行性而进行的一项试验工程。

1972 年在首届国际计算机通信会议(ICCC)上首次公开展示了 ARPANET 的远程分组交换技术。分组交换不同于传统电信网中采用的电路交换,是**存储转发交换**方式中的一种交换方式,它将要传送的报文分割成许多具有统一格式的分组,并以此为传输的基本单元,一一进行存储转发的传输。和电路交换相比,分组交换具有线路利用率高、可进行数据速率的转换、不易引起堵塞以及具有优先权使用等优点。因此,它被广泛用于计算机网络。

1974 年 IBM 公司首先公布了系统网络体系结构(SNA)作为 IBM 计算机的连网标准。之后,各大计算机厂商都相继开发了自己的网络体系结构,如 DEC 公司的数字网络体系结构(DNA)等。

1976 年国际电报电话咨询委员会(CCITT)制定

了用于公用分组交换网的协议标准 X.25,推动了公用分组交换网的发展。

1976 年美国 Xerox 公司开发了基于带碰撞检测的载波监听多路访问(CSMA/CD)原理,用**同轴电缆**连接多台计算机的局域网,取名为**以太网**。由于以太网安装使用方便,性能较好,成为最广泛使用的一种**局域网**。随着个人计算机(PC)的广泛使用,局域网的研究、开发和应用有了很大的发展。

1978 年国际标准化组织(ISO)为了解决不同厂商的计算机网之间不能互联的问题,提出了**开放系统互联基准(参考)模型**(OSI/RM),即 OSI 网络体系结构,以推动网络标准化工作。

在总结最初的建网实践基础上,DARPA 组织有关专家开发了 ARPANET 第三代网络协议——TCP/IP(参见 **TCP/IP 协议集**),并于 1983 年在 ARPANET 上正式启用。与此同时 UNIX BSD 版安装了 TCP/IP 协议软件。

目前采用 TCP/IP 的 Internet 已经成为全球最大的、开放的、由众多网络互联而成的计算机网络。Internet 的发展已经历了三个阶段,逐渐走向成熟。从 1969 年 Internet 的前身 ARPANET 的诞生到 1983 年是 Internet 的研究试验阶段,它主要是作为对网络技术进行研究和试验用的网络。从 1983 年到 1994 年是 Internet 的实用阶段,作为用于教学、科研和通信的学术网络在美国和一部分发达国家的大学和研究部门中得到广泛应用。从 1994 年以后,Internet 开始进入商业化阶段,除了原有的学术网络应用外,政府部门、商业企业以及个人广泛使用 Internet,全世界绝大部分国家都已经接入 Internet,成为全世界的信息基础设施。

当前使用的 Internet 的协议标准为 IPv4(第四版互联网协议),只有约四十亿个地址(2^{32}),已经于 2011 年分配完毕。下一代的 Internet 的协议标准为 IPv6(第六版互联网协议),有 2^{128} 个地址。由于 IPv4 和 IPv6 并不兼容,目前正在处于从 IPv4 向 IPv6 的过渡阶段。

由于 Internet 的社会需求飞速增长,它正面临着多种挑战,包括网络的带宽和可扩展性、网络的安全性、网络的服务质量、多种新的网络应用需求以及引发的商业、文化和社会问题。在学术界,面向十年以后的未来网络体系结构的研究也在进行之中。

网络分类

计算机网络分类方式繁多,一般有以下几种:

(1) 按地域范围可分为**局域网**、**城域网**和**广域网**等;

(2) 按拓扑结构可分为总线、星状、环状、网状等网络;

(3) 按交换方式可分为**电路交换网**、**分组交换网**等;

(4) 按网络协议可分为采用 TCP/IP、SNA、SPX/IPX、AppleTALK 等协议的网络;

(5) 按安全控制政策可分为内联网、外联网等。

基本内容

随着计算机网络应用的日益广泛,网络规模的日益扩大,都对网络硬件和软件提出了新的要求,因此,计算机网络技术也得到了高速的发展。网络主要技术包括网络体系结构、网络互联、数据通信、局域网、承载网、网络管理、网络安全与泛在网等。

网络体系结构

网络体系结构是计算机之间相互通信的层次以及各层次中的协议和层次之间接口的集合。**网络协议**是计算机网络和分布系统中相互通信的对等实体间交换信息所必须遵守的规则的集合。采用 **TCP/IP 协议集**的网络体系结构是计算机网络体系结构的主流,并仍在改进和发展中。**网络协议工程**是一门研究如何设计和构造协议规范以及如何把所设计和构造的协议规范快速、准确、低成本地转化为可执行代码的工程。

网络互联

网络互联将多个网络互相连接以实现在更大范围内的信息交换、资源共享和协同工作。Internet 就是由成千上万个不同的网络互联而构成的网际网,或称互联网。**网络互联协议**是计算机网络间互相连接进行通信时有关数据格式及交互过程必须遵循的约定。IP 是一种最著名的网络互联协议,由 Internet 广泛采用(参见**网际协议**)。**路由选择**是在网络环境中寻找一条到达目标计算机通路的过程。实现路由选择的算法称路由选择算法,并有相应的路由选择协议。常用的**网络互联设备**有**交换机**、**路由器**、**网关**。

数据通信

自 20 世纪 80 年代以来计算机科学和**数据通信**的融合,大大改变了这些领域的技术和产品,推动了计算机-通信产业的发展。数据通信是依据指定的规程或协议,将数据源的数据编码,通过传输介质发送到目的地的过程和技术。

局域网

局域网是把位于局部区域内多台计算机等终端设备通过传输介质和通信设备互连起来的数据通信

网络,通常覆盖距离小于几千米和几十千米。通信设备包括网桥、交换机和路由器等。通信传输介质分为有线和无线两种,如双绞线和微波。数据传输速率从早期的 1Mbps 和 10Mbps 逐步发展到 100Mbps、1Gbps、10Gbps 和 40/100Gbps 不等。在局域网内,通过高层协议和功能完善的网络软件,可以实现计算机等终端设备之间的相互通信和资源共享。

承载网络

为了形成连接若干个城市、地区,甚至跨越国界,遍及全球的一种计算机网络,需要使用通信公司提供的承载网。公用分组交换网(X.25),帧中继(Frame Relay)和**异步传送模式**(ATM)是早期的承载网。随着互联网的普及,承载网逐步演化为主要提供点对点的通信信道。近年来,光通信技术的很快发展,推动了承载网的发展,包括**波分复用**,**光交换网络**等。此外,**移动通信网**、**卫星通信网**也是承载网的重要组成部分。

泛在网

泛在网并没有严格的定义,通常被理解为是一种广泛存在的网络,它可以在任何时间、任何地点、为任何人及任何物提供顺畅的通信联系,通过合适的终端设备与网络进行连接,获取个性化信息服务。

网络管理

网络管理功能包括对网络的配置、故障、性能、安全、计费等进行管理的功能。**OSI 管理体系结构**是基于开放系统互连环境对资源进行管理的一种体系结构,它提供了在开放系统互连环境中控制、协调和监视各种资源的手段。**简单网络管理协议**(SNMP)是基于 TCP/IP 协议的一种功能比较简单的网络管理协议,被广泛用于 Internet。**公共管理信息协议**(CMIP)是由国际标准化组织(ISO)为开放系统互连(OSI)制定的网络管理协议标准。

网络安全

网络安全是在分布式计算环境中,对信息的传输、存储、访问提供安全保护,以防止信息被窃取、篡改和非法操作。网络的安全问题来自两个方面,一个是网络内部的安全漏洞,另一个是网络外部的安全威胁。网络安全的三个基本要素是保密性、完整性和可用性服务。在分布网络环境下还应提供**鉴别**、**访问控制**和抗否认等服务。完整的信息安全保障体系应包括考虑到网络内部安全漏洞和网络外部安全威胁所涉及的保护、检测、响应、恢复四个方面。常用的安全防范技术包括身份鉴别、访问控制、完整性控制、密码技术、**防火墙**系统、**计算机病毒**保护、审计和恢复、**操作系统**安全、**数据库**系统安全等。

计算机网络发展趋势

计算机网络发展的总体目标就是要在各个国家,进而在全球建立完善的信息基础设施。信息基础设施将改变人们的生活、学习、工作、人际交往的方式,提高人民的生活质量,推动社会的进步。

1993 年美国政府制定了信息高速公路发展计划后,各国政府都相继规划和实施相关的计划。这些国家级网络的建设目标是在全国范围内建立为民众普遍服务的信息基础设施。进入 21 世纪以来,商业化互联网的发展使社会迅速进入了信息化的阶段。新的网络应用层出不穷,极大促进了社会经济和文化的发展,也在各个方面带来了巨大的挑战。

计算机网络有三个重要的支撑技术,即微电子技术、光传输技术和无线通信技术。

微电子技术的发展是信息产业发展的基础,也是驱动信息革命的基础。其发展速度是惊人的,用摩尔定理来预测,微电子芯片的元件数量每 18 个月提高一倍。

驱动信息革命的另一个支撑技术是**光通信技术**。评价光纤传输发展的标准是传输的比特率和信号在需要再生前可传输的距离的乘积。在过去 10 年间,该性能每年增长一倍,这种速度可望再持续 10 年到 15 年。

随着移动网络的到来,**无线通信技术**有了长足性的进展。随着增加频谱,采用数字调制,改进编码技术和建立微小区和宏小区,无线系统的容量将增加 1000 倍以上。

计算机网络技术发展的重要特点是融合,包括计算机、通信、信息内容这三个方面的融合,包括电信网、电视网、计算机网的融合也包括固定通信网和移动通信网的融合。这些融合的特点包括数字化和分组化。人类正快速进入数字世界,所有通信方式,包括话音、图像和数据都能够而且都将被数字化,这样信息之间的转换、重组、存储、处理、传输都变得十分简易,为各种信息交互的模式提供了基础。计算机网络体系的核心是分组化,“IP over every thing”和“every thing on IP”这两句话很好地概括了 IP(参见**网际协议**)的重要特征。以 IP 为核心的网络体系结构,向下可兼容各种不同的通信机制,向上可实现各种不同的业务。

计算机网络目前的热点有 IPv6、移动互联网、云计算、物联网、网络安全和未来互联网/未来网络等。

IPv6 可以解决网络地址的可扩展问题；**移动互联网**使用无线技术，可以随时随地联网；**云计算**使联网用户可以使用计算资源和存储资源；**物联网**使物理世界和信息空间产生交互；**网络安全**是网络信息基础设施和用户隐私保护的基础；**未来互联网/未来网络**是研究并设计新一代的网络。

参考文献

Tanenbaum A S, Wetherall D J. 计算机网络. 5版. 严伟，潘爱民，译. 北京：清华大学出版社，2012

（李星　胡道元）

jisuanji weihu

计算机维护（computer maintenance）　为保证计算机系统正常、高效运行而进行的各种调整、检测、监控、维修及保养等项工作。计算机维护是计算机投入运行后必须进行的日常工作。它对提高计算机系统的运行效率，保证数据（信息）的安全与完整，延长计算机设备的使用寿命都有十分重要的作用。

计算机系统的运行维护　计算机系统安装调试完成后，一旦投入运行，系统维护即成为不可缺少的日常工作。系统运行维护的目的是为了使系统更有效地运行并保证系统中信息的安全与完整。

资源利用监控　系统维护人员必须经常监视计算机资源（如**中央处理器**、**主存储器**、**输入输出通道**等）的运用情况并进行必要的均衡与调整，以使有限的资源得到最大限度的利用。某些情况下仍需要人工进行干预，以均衡不同作业对资源的利用，消除影响系统高效运行的各种"瓶颈"。由于多个作业争夺资源出现的"死锁"，或某些作业由于程序缺陷而出现的无休止循环，都会造成资源的浪费，均需要人工干预予以排除。

作业管理　对多用户多任务的应用系统，作业管理的任务包括根据作业的需要为其分配资源并适时予以回收；根据待处理作业的不同类型，对资源的不同需求以及当前运行情况进行作业的调配；进行分时时间片的设置以及定时启动或中止某些作业的运行；给定与调整不同类型作业的优先级，如白天给予交互式作业高的优先级，晚间与假日则将处理机资源主要分配给大型批处理作业等。

信息转储　系统运行过程中不断产生大量信息，包括系统本身的运行状态信息及在系统中运行的各种作业信息。为了保证信息的安全性与完整性，也为了充分发挥高速存储器（主存储器及**磁盘存储器**）资源的作用，各类应用系统都必须根据具体情况，从主存储器或磁盘存储器向磁带或U盘、光碟等介质转储系统运行的状态信息及作业的有关信息。对大型连续运行的应用系统，转储的频度至少每天一次。即使是**微型计算机**系统，也需经常进行信息转储，例如在磁盘空间将用满或一些不常用的文件需要保存的情况，就应将盘空间的信息转储到U盘、移动硬盘或光碟。转储的介质应存放于安全处所，对特别重要的转储信息，应有两份拷贝存放于不同地点，以防止因系统故障及其他突发事件导致大量信息丢失而造成严重的后果。

用户服务　对于大型系统，用户服务是一项不可缺少的维护工作。如用户的登录与注销；接受用户的咨询，使用户了解人机之间的界面；根据用户的请求以相应介质输入、转储或输出信息，并为用户保存信息存储介质；通过系统控制台与用户通信，在系统发生突然情况时向用户通告，以便采取必要措施妥善处理作业的异常中止等。

软件维护　**软件维护**是系统维护的一项重要内容。系统投入运行一段时间后，硬件初期易发故障已经排除，性能趋于稳定，出现故障相对较少，软件维护就成为系统维护的主要工作，例如系统软件及应用软件隐含的某些缺陷需在一定条件下暴露后才能加以排除，又如各种软件的性能升级及版本更新等都是经常需要进行的工作。

元器件的预防性维护　计算机是由大量光、机、电元器件集合组成的装置，元器件数量巨大且通常处于连续运行的状态，不可避免地存在设备老化及寿命问题。为了尽可能降低运行中出现设备故障的概率，提高系统的可靠性，应采取以预防为主的维护措施，对偏离正常状态或接近使用期限的元器件、零部件重新调整其工作状态或予以更换，以保证系统连续正常地运行。

系统初始启动的例行检测　一个准备投入运行的系统，在系统加电后首先运行基本测试程序，检查系统各个部分是否正常工作，如未发现异常即可进入运行状态，否则即停机检修后再投入运行。较先进的系统在运行中间，其诊断部件会自动对系统中的关键部件进行巡回检测，并对检测结果予以记录或显示，系统维护人员通过监视检测情况可及时了解运行状态，必要时采取措施防止错误发生或扩大。

定期进行系统的全面检测　对较复杂的计算机系统，应定期运行系统的综合性诊断检测程序，并分析近期的运行记录，判断系统是否存在隐患，是否发生

过偶发性故障,某些部件的组成部分是否已被自动切换。目前较先进的计算机系统均采用容错技术以提高可靠性,如通过复制手段纠正逻辑操作的偶发性错误;通过校验手段纠正存储器的单个位错误;对某些功能相同的寄存器及存储器阵列设置备份,需要时进行自动切换等。这些系统自动进行的维护措施会在系统的运行记录中有所反映,维护人员根据综合测试及分析运行记录的结果采取更换设备或重新配置等措施,使系统保持在良好的工作状态。一般大中型系统,每周应进行 1 次 ~2 次全面检测,小型微型系统可适当延长全面检测的时间间隔。

定期进行机电设备的维护保养　对易损易污的机电运转部件(如**磁带机**、打印机等)、通风过滤装置进行清洁、润滑或更换零配件;调整敏感元器件的阈值电压及基准电平;调整精密装置如软磁盘机的读写磁头运动机构相对于盘片零道的位置;调整真空部位(如磁盘机、磁带机的气动部位)的气压等。上述设备的维护保养对种类繁多结构复杂的外围设备尤其重要。外围设备是计算机系统发生故障最多的部分,做好设备的维护保养可大大降低整个系统的故障率。一般应一个月进行一次局部的机电设备维护保养,一季度至半年进行一次全面的机电设备零配件更新与维修。易损易污部件的清洁工作(如磁带机读写磁头沾染的磁粉擦抹,打印机走纸机构的纸屑清除等)则应每天进行。

计算机发生故障后的临时性维修　如计算机发生故障导致系统不能运行则应停机进行临时性维修。首先要区分是软件故障还是设备故障。软件故障可能是因为系统软件的某个环节在特定组合条件下不能正常运行引起的,也可能是多种作业在运行中因争夺资源而出现死锁等原因造成的。这类故障一般可采用重新启动系统或其他人工干预手段予以排除。如果是设备性能变坏引起的硬件故障,则应使系统从运行状态转为故障诊断状态,运行诊断测试程序检测全机各部件,特别是**中央处理器**和**磁盘存储器**两种部件(输入输出部件一般不至于影响整个系统的正常运行),尽快故障定位,然后针对故障部位进行维修。

中央处理器的临时性维修　中央处理器的故障原因主要是集成电路失效。当前的计算机系统均配备较完善的诊断测试手段,提供详细的故障维修指南,对大部分电路的故障可以实现准确定位。但由于集成电路组装密度很高,一个集成电路芯片包含的逻辑单元或存储单元数以百万计,诊断测试程序检测出的故障通常定位于一个电路模块或一个乃至几个电路卡,维护人员根据测试结果能在现场进行的维修工作就是更换电路卡。如现场没有相应的备份件,可以采取降级运行(如多处理器系统可切除有故障的处理器,存储器可切除部分有故障的单元等)的手段使系统保持连续运行,若没有补救手段则需停机检修。

外围设备的临时性维修　对外围设备的故障检测应采用脱机检测与联机检测两种方式。脱机检测是指外围设备在逻辑上与中央处理器脱离联系(必要时把物理连接也脱离)的情况下对不同外围设备运行特定的测试程序,进行不含接口部分的功能测试,借助设备的面板或专用测试器所显示的信息并参阅维修手册判断故障所在部位。外围设备的故障有一类是集成电路失效,可通过更换电路卡排除;另一类是各种外设的特殊故障,如磁盘盘面损伤、读写磁头位置偏离或其运载机构不能正常运动、打印机的打印部位损坏或打印纸传递机构故障等,需根据具体情况进行维修。如脱机测试正常而联机却不能正常运行,则应进行针对该设备的联机测试,运行相应的测试程序,测试该设备与中央处理器的接口部位并检验两者之间的协调关系;必要时还可进行模拟环路测试,即将外围设备至主机之间的输入输出连线构成回路,以确认故障所在部位是否在接口电路。

电源部件的维修　计算机系统的各个部分均有专用的电源部件,电源部件中有一部分是大功率的分立器件,故障率较高,是常见的故障部位。系统维护时除了定期对这部分元器件进行检修更换外,在检测中央处理器及各种外围设备时,如发现工作异常,应充分注意到电源部件是可能发生故障的主要部位。

随着计算机技术的发展及各种应用对计算机系统可靠性、可用性不断提高的要求,今后的计算机系统将更广泛采用冗余技术,如动态重配置、磁盘镜像等技术,尽可能避免由于硬件故障而引起的系统运行中断。这样既提高了系统的可靠性和可用性,也降低了临时性维修的难度。　　（苏振泽）

jisuanji xitong

计算机系统(computer systems)　由一台或多台计算机以及相关软件组成的功能单元。该功能单元:①使用公共存储设备来存储一个程序的全部或者部分以及执行该程序所需的数据的全部或者部

分；②执行用户编写的或用户指定的程序；③执行用户指定的数据运算，包括算术和逻辑操作。一个计算机系统可以是单个独立的系统，也可以由几个相互关联的系统组成。

从用途上看，用于科学与工程的计算机，通常需要很高的性能，因此产生了高性能计算系统（参见**高性能计算**），计算结点通过高带宽、低延迟的**互连网络**连接起来，协同解决重大的科学和工程问题。高性能计算系统主要通过并行处理的方式（参见**并行处理系统**）来提高计算机系统的性能。**互联网**的兴起、分布式操作系统和分布式数据库等的发展，使得**分布式处理系统**得到越来越广泛的应用。分布式处理系统通过相对松散耦合的网络将单个计算机系统连接起来，可以提高系统的性能和可靠性。

从应用方式来看，网络技术提供了计算机之间高性能、低成本的通信方式，可以将不同的计算机连接起来，从而催生了一些新的应用模式。例如，**云计算**通过对资源的共享和统一管理，实现了对资源的按需获取，可以提高计算机系统的使用效率和灵活性，并降低成本。**网格计算**通过将若干高性能计算机系统连接起来，可以完成过去很难由单个系统完成的复杂任务。**对等计算**通过去中心化、在用户之间直接共享数据和计算资源，提高了系统的容错性和传输性能。另一方面，芯片技术的进步使得计算机体积可以越来越小，并可以随着人或交通工具移动，产生了**移动计算**。芯片的进一步小型化和微型化使得计算机无处不在，催生了**普适计算**模式。

（陈文光）

jisuanji xitong RAS jishu

计算机系统 RAS 技术（computer system reliability, availability and serviceability） 计算机系统可靠性、可用性和可服务性 3 项技术的总称。它是研究、设计、评价计算机系统，特别是高可靠的计算机系统必不可少的重要内容，是系统高可靠、高可用和高可服务的重要技术手段。系统的 RAS 能力已成为衡量大中型计算机以及服务器系统综合性能的一个重要技术指标。

发展简史

本来广义的计算机系统可靠性就包含可用性和可服务性在内。20 世纪 70 年代后期，由于可用性和可服务性技术的发展，才产生了更为直观的 RAS 概念，突出了人们对可用性技术和可服务性技术的关注。

计算机系统可靠性技术是与计算机系统本身的发展同步的。1951 年，第一台商品计算机 UNIVAC Ⅰ就采用了奇偶检验码，并用两个运算部件，以匹配-比较方式工作。这个时期计算机系统的 RAS 特性基本上取决于事后的功能测试（即软件方法）；硬件的错误控制只赋予个别的奇偶校验或比较功能；而且所有故障的处理都是由人工实现，即纯经验型的维护方式，因此不具备组织 RAS 系统的条件。

1969 年 STAR 容错计算机研制成功，进一步促进了容错技术的发展。这个时期计算机发展到第三代，由于采用集成电路，系统的可靠性有较大幅度的提高，同时由于引入了微程序控制方式，实现了对系统的半功能测试（微诊断方法），使 RAS 技术渗入计算机系统的结构中（如故障诊断技术、指令及微指令复执、以字节为单位的奇偶校验、1 比特错误的自动纠正等容错技术），但这些技术在计算机系统设计时是作为部件或系统的附加功能而个别引用的，系统的诊断硬核仍是主系统本身。这一代计算机所提供的丰富的 RAS 功能，为组织 RAS 系统准备了必要条件。

随着大型计算机系统的发展，特别是由于 20 世纪 80 年代超大规模集成电路的发展和计算机应用的普及，人们对计算机 RAS 技术性能的要求愈来愈高。这个时期计算机发展到第四代，在计算机系统设计中，可靠性、可用性、可服务性和经济性等基本参量得到权衡，以保证足够的 RAS 功能为前提，整个系统的性能价格比得到了大幅度的提高。计算机系统开始普遍采用独立于主系统的维护子系统，从而使诊断硬核移出主系统，并建立了以 RAS 功能为目的的软、硬件综合体——RAS 系统。

计算机 RAS 技术在军事、航天、金融等对计算机可靠性有苛刻要求部门应用广泛。对于实用的计算机系统，由于系统规模、应用场合、环境条件的差别以及成本价格因素的影响，其 RAS 技术的投入强度和实现方法是有所不同的。

基本内容

1. 可靠性

系统在规定的工作条件下和规定的时间内持续正确运行（完成规定功能）的概率。可靠性通常用**平均故障间隔时间**（MTBF）来表征（参见**计算机系统可靠性**）。

在进行系统可靠性设计中所采用的可靠性基本技术主要包括：

（1）系统可靠性预计　计算机系统由各类元件组成。系统可靠性预计即根据各类元件的可靠性以及各元件之间的连接关系构成的可靠性模型做系统可靠性的预先分析计算，以预测系统是否达到可靠性指标；找出关键元件、关键路径，为改进可靠性设计提供依据。

（2）可靠性分配　根据系统总的可靠性指标，将其分解，并对各分系统，直至器件、工艺提出相应的可靠性指标。

（3）元件优选与筛选　在成本允许范围内，选择高可靠元件，并进行老化筛选。

（4）线路工程　通过对关键线路的工程试验与分析，制定工程设计规范，确定生产工艺参数，将外部干扰和内部噪声控制在容限范围内。

（5）热设计　合理布局热源，制定冷却方案，控制元器件工作环境的温度和湿度。

（6）容错设计　通过冗余技术消除或减轻机器校验出错或故障造成的影响。冗余技术包括硬件冗余、软件冗余和时间冗余。硬件冗余有校验纠错、双工系统、多数表决和其他复式冗余结构。软件冗余有关键数据和程序的重复存储，设置多个程序入出口等。时间冗余主要以时间为代价，通过重复执行（简称复执）消除偶发性故障的影响。复执可以在微操作、指令、程序段或整个作业等各种级别上进行。多数计算机系统往往同时采用上述多种冗余技术。

2. 可用性

系统在规定的工作条件下能正常工作的概率。一般用平均利用率 A 来表示，即

$$A = \mathrm{MTBF}/(\mathrm{MTBF} + \mathrm{MTTR})$$

式中，MTTR 为系统的**平均恢复时间**；MTBF 为平均故障间隔时间。

如果把系统正确完成规定功能的总时间称为正常运行时间，把系统维修总时间称为故障时间，则平均利用率 A 可表示为

$$A = 正常运行时间/(正常运行时间 + 故障时间)$$

通常把 A 叫作系统可用率。

可用性技术是基于可靠性技术和可维护性技术，旨在提高系统可用性所采用的技术。主要包括：

（1）冗余结构　在系统构成上采用复式模块化冗余结构。当某一部件出现故障时，可自动或人工重构系统，由另一部件替代故障部件工作，系统降级使用，除故障部件外，其他部件仍保持正常运行。

（2）并行维修　为故障部件建立维修环境，使维修工作与系统正常运行同时进行。

（3）容错计算　通过多重处理或其他软硬件相结合的技术手段，消除机器故障造成的影响，使用户感觉不到机器曾经发生过故障。

（4）热替换　许多系统均支持部件的在线更换，包括磁盘、风扇、电源等的在线更换、PCI 卡的热插拔，某些系统还支持在不停机的情况下对组件或模块进行替换。

（5）数据完整性保证　数据的完整性保证体现在系统硬件的各个层次，与系统硬件设计和采取的策略密切相关。中央处理器（CPU）接口、高速缓冲存储器（cache）、存储器接口通常采用奇偶校验和 ECC 检错纠错方式；印制板（PCB）内部器件之间和输入输出（I/O）总线采用奇偶校验；互连通信接口采用 ECC 或 CRC 检验保证数据传输的正确性。

（6）错误恢复　错误恢复是指在系统硬件出现故障时，系统能够自动隔离故障，保证系统正常运行，或者在系统崩溃后能迅速隔离故障、重启系统。对错误的处理一般通过硬件和软件协调进行，对可恢复的错误进行纠错，对不可恢复的错误则终止程序或重启系统。

3. 可服务性

系统可服务性是系统可维护性和可修理性的总称。可维护性反映了计算机系统维护及排除故障的性能（参见**计算机系统可维护性**）。可修理性是指系统发生故障后，从查找故障原因到更换故障部件或修改故障软件使之重新投入正常运行的支持能力。一般，可服务性用**平均修复时间**（MTTR）来表征。

可服务性技术主要包括：

（1）系统控制台　集中监视机器运行状态，校错，故障报警，提供必要的软件、硬件维护以及调试支持。

（2）自测试　在机器建立初始环境或在正常运行过程中，自动进行功能测试。

（3）故障自动诊断　一旦机器发生故障，即自动或人工进入故障诊断，定位故障点。

（4）连机维修　允许在正常运行或带电状态下更换故障部件。

（5）维修培训　使维修人员熟练掌握维修技术，缩短维修时间。

计算机系统应提供有效的管理、监控和诊断子系统，由系统监控和管理服务器以及相应的硬件诊断和监控网络构成，对主机进行管理和监控，完成

系统分区、运行时诊断和日志登记等功能，并提供高级诊断支持。

综上所述，可靠性、可用性、可服务性3项技术各有其自身的技术含义，又相互关联。上述平均利用率的表示集中体现它们之间量的关系。

发展趋势

良好的RAS性能是计算机系统高性能的重要保障，随着传统的RAS技术的成熟，新的RAS技术和方法也相继出现，包括以下方面。①数据完整性的新技术研究，如提供多位验错及校错功能，在内存故障时对数据进行透明恢复。②芯片级RAS技术，随着芯片技术发展和工艺水平的提高，单芯片功能和规模将越来越强大和复杂，芯片级可以提高系统的整体RAS能力，并减低系统在RAS方面的成本。③标准协议内的RAS技术，在计算机系统各部件之间以及与I/O设备之间连接的标准协议采用RAS技术，有利于提升整合系统的RAS能力。④高可用系统设计，方法之一是采用冗余部件，对一些关键部件添加冗余配置，设计方便的更换结构（如热插拔），并对一些关键的数据通路提供冗余路径；方法之二是实现故障在线恢复与系统重构，系统能够实时检测故障并进行故障恢复，实现系统的动态升降级。⑤灾难恢复，采用备份或镜像的方法，包括数据备份和系统备份、磁盘镜像、单节点系统镜像等，或采用软件检查点，包括高效的、对用户透明的系统级检查点/重启机制。⑥故障预测与故障诊断的智能化，及时了解系统的运行状态和可能的故障点，提醒用户及时调整系统应用策略，增强系统的可用性和可维护性，并可预先更换可能的故障部件，保证系统的可靠运行；⑦嵌入式专用部件，计算机内部的专用嵌入式系统，用于计算机的管理控制以及对电压、温度和风扇的实时监控和实时处理，并为计算机系统提供远程维护功能，这成为提高系统可用性和可维护性的重要技术手段之一。⑧专用RAS集成支撑环境，采用分布式集中管理的形式，结合嵌入式专用部件，形成单一的、易于管理的RAS网络，将系统的监控、管理、诊断、维护等功能集成为专用RAS支撑环境，其规模将具有可扩展性，以适应系统规模的可扩展，这种RAS集成支撑环境还可以对系统可用资源进行调配和管理，与操作系统相结合实现系统的动态升降级和系统重构等功能。

参考文献

1. 傅佩琛，等. 计算机系统硬件软件可靠性理论及其应用. 北京：国防工业出版社，1990

2. 蒋句平，庞征斌，周兴铭. 高性能计算机RAS技术现状与趋势. 计算机工程与科学，2005，第1期

3. 汪文华. 现代计算机结构中的RAS系统. 武汉大学学报（自然科学版），1987，第1期

（羌卫中　陆绍福）

jisuanji xitong kekaoxing

计算机系统可靠性（reliabiliy of computer system）　在规定的条件下和规定的时间间隔内，计算机系统能正确运行的概率。“规定的条件”包括环境、使用、维修等条件和操作技术；“规定的时间”是指可靠性是对一定的时间间隔而言，人们常常要求能在规定的时间内具有一定的可靠性；“正确运行”是指系统能完成规定的各项技术性能，运行的程序不被破坏或停止，程序按规定的要求运行，运行结果正确，执行时间不超过一定限度等。

基本内容

计算机系统可靠性通常用**平均故障间隔时间（MTBF）**，即系统能正确运行时间的平均值来表征。

系统可靠性一直是计算机的一项重要指标。随着计算机应用范围的扩大，要求计算机系统不但有良好的技术性能，同时还必须有较高的可靠性。提高系统可靠性一般有两类技术方法，即避错法和容错法。

硬件避错技术的作用是减少失效的可能性。主要有：①对元器件进行老化筛选。②使用可靠的连接组装技术，严格工艺生产过程中的质量控制。③在设计时对元器件的额定参数留有足够余量，即电应力、热应力的实际使用值明显地低于额定值，通常只有额定值的几分之一。④由于元器件参数的离散性和负载变化产生的参数漂移，在基本节拍内逻辑链级数延时的设计留有足够余地。⑤降低系统内部的电磁干扰。如印制电路板中，限制平行走线、重叠走线、分叉走线的长度和数量；限制接插件上同时同相动作信号线的数量；采取地线和地平面、电源线和电源平面等隔离技术以及电源的高、低频滤波等。⑥高速信号线采用传输线及匹配技术。⑦屏蔽外界电磁干扰。⑧做好热设计，必要时采取冷却措施（风冷、水冷），降低机柜内的温度。⑨采取防震、防冲击、防潮、防盐雾等机械结构措施。通过上述避错技术可使系统能承受一定的工作条件变化，如电源拉偏、频率拉偏、温度变化、电磁干扰、机械振动等。

提高可靠性的另一种方法是容错法，主要采用

硬件冗余、软件冗余、信息冗余和时间冗余等技术,屏蔽故障的影响。使系统出现故障时,仍能保持正常的功能。

硬件容错技术包括故障诊断、故障屏蔽和动态冗余技术。故障诊断是检测系统故障的发生及其确切位置,它是实现故障屏蔽和动态冗余的先决条件。故障屏蔽是一种静态硬件冗余技术,它通过冗余资源来隔离或校正故障的影响,如双工冗余、模3表决冗余、海明码纠单错等。屏蔽技术只是容忍一定程度的故障,当冗余资源耗尽后,再发生故障,系统就会产生错误的结果。动态冗余是综合性的容错技术,已广泛应用于容错计算机系统中,其基本思想是:在发生故障时,通过系统内部的自动重组来切除故障部件,并以备用部件替换故障部件。实现动态冗余,要求系统具有模块化结构和故障检测、定位功能,用检测到的故障信号激活故障处理程序,自动将故障部件处理的任务转移到完好的部件或备用部件,并启动系统运行或性能降级运行。故障部件可以联机维修或脱机维修(使用活线拔插技术),修复后再加入系统,不中断系统运行。动态冗余有电路级、分系统级和系统级冗余:电路级冗余指在系统电路的关键部分设置备份电路,在发生故障后启用;分系统级冗余是指分系统装置多重化,如采用冗余多处理器系统,可互为备份或采取服从多数裁决的策略;其他如输入输出部件、存储部件、系统互联网等都可设置冗余。系统级冗余则有双工系统、均分负载系统和双机系统等。

时间冗余又称“复执”,通过重复执行发生故障的操作,使由于某种不明原因而引起的瞬时偶发性故障不再出现。复执有硬件复执和软件复执两种:硬件复执由增设的控制电路,在出现故障后,延迟若干时间,重复执行发生故障的操作,如此时偶发性故障业已消失,则复执结果可获成功,延迟的时间长短和1次故障需要复执的次数可根据经验设定;软件复执可在单指令、1段程序或整个作业3个层次上进行,复执方案根据瞬时偶发性故障的多少、程序的检查点设置、复执所付出的代价以及复执的成功率等来决定。

为了保证系统的可靠性,在系统的设计阶段就应将可靠性作为一项重要的设计内容,除了采用上述避错和容错技术方法外,还应对系统进行可靠性分配与可靠性预计。

可靠性分配是根据用户对计算机系统可靠性指标的要求,对各分系统、设备等各部分提出相应的可靠性指标,并逐级分配到集成电路、元器件、接插件、印制电路板、生产工艺质量等项目上。在实际设计中,一般要提出多个方案进行比较、调整,以求得合理的结果。

可靠性预计是对系统的可靠性进行估算,简单的计算方法是先求出分系统的失效率 λ_i

$$\lambda_i = m_1\lambda_{i1} + m_2\lambda_{i2} + \cdots + m_K\lambda_{iK}$$

式中,K 为组成第 i 个分系统的成分种类,如集成电路、元器件、接插件、印制板金属化孔、焊点等;m 为各成分的数量,$m_K\lambda_{iK}$ 为第 K 种成分(例如焊点)的失效率。

再求出系统的失效率 λ

$$\lambda = n_1\lambda_1 + n_2\lambda_2 + \cdots + n_L\lambda_L$$

式中,L 为分系统(部件,设备)的种类;n 为分系统的数量;$n_L\lambda_L$ 为第 L 中分系统的失效率,则系统的平均故障间隔时间 MTBF 为

$$\mathrm{MTBF} = \frac{1}{\lambda}$$

MTBF 的单位为 h(小时)。在实际使用时,一般还要考虑:①系统所处的环境;②反映设计成熟程度的修正因子;③反映分系统冗余程度和容错能力的加权因子等。预测的失效率与实际的故障率有一些差异是正常的,设计阶段的可靠性预计只是一种估算。

计算机系统可靠性技术是计算机科学与工程的一个重要领域。由于超大规模集成电路的发展,出现了各种类型高性能计算机。计算机互连和数据传送的高可靠性技术和计算机系统的其他高可靠性措施都将有很大的发展。系统的正确性证明、系统故障的自诊断、自修复技术、软件容错技术和系统的高保密性、新的防病毒措施等,将取得进一步发展。随着对 Petascale/Exascale 系统的研究,传统的容错技术与方法(如复执技术、动态冗余技术)面临代价过高的问题,错误避免、错误预测等技术手段被广泛研究。

参考文献

1. Avi zienis A, Laprie J C, Randell B, et al. Basic concepts and taxonomy of dependable and secure computing. IEEE Trans. on Dependable and Secure Computing, 2004, 1(1)

2. 傅佩琛,赵霖,张军英. 计算机系统硬软件可靠性理论及其应用. 北京:国防工业出版社,1990

3. 朗博顿. 计算机系统可靠性. 张复,吴仲贤,译. 游鄂毓,校. 北京:国防工业出版社,1988

(石宣化　陈玉霜　应秋丽)

jisuanji xitong keweihuxing

计算机系统可维护性(maintainability of computer system) 计算机系统保持或被修复到执行规定功能状态的能力。计算机可维护是计算机系统持续正常运行的一个重要保障。

可维护性的概念最早是由 Alan Stewart Goldman 和 T. B. Slattery 在 1964 年给出的。衡量计算机可维护性的指标称为可维护度,记为 $M(t)$。一个计算机系统的可维护度 $M(t)$ 是指该系统失效后,在规定的时间间隔 t 内被修复的概率。在系统的失修率和修复率都是常数的情况下,根据可维护度的定义,可得可维护度 $M(t)$ 与修复率 μ 的关系为

$$M(t) = 1 - e^{-\mu t}$$

修复率表示在单位时间内完成修复的概率,亦即反映系统可维护性高低的一个常数,系统可维护性的另一个重要参数是**平均修复时间**(MTTR)。它是指对失效的系统进行修复所需的平均时间。平均修复时间与修复率的关系为

$$\text{MTTR} = \frac{1}{\mu}$$

可见,修复率 μ 对于一个系统的可维护性至关重要。修复率除了依赖于修复人员的技术水平外,还依赖于修复的环境和条件。

平均修复时间 MTTR 可用实验方法求得:对一个系统注入不同的故障,每注入 1 个故障就进行修复并记下修复时间,最后可求出平均修复时间 MTTR。为了使所求得的 MTTR 比较准确,在进行实验时应注入足够多的不同类型的故障,并让不同技术水平的维修人员进行修理。

在实际工作中,一般将维修分成 3 级。

第一级维修为现场级,包括所有能在系统所在地(如计算机房)进行的维修。通常现场级维修要求将故障定位至电路板级,然后更换有故障的电路板,从而使系统恢复正常运行。由于不可能将复杂的故障检测与定位仪器带到现场,现场级维修通常要借助于系统所提供的内装测试功能。

第二级维修为中间级。在现场不能实施的维修可在临近现场的维修点进行,称为中间级维修。通常在维修点具有比在现场更好的维修设备及更充足的备件,因此中间级维修能解决在现场不能解决的问题。另一方面,由于维修点比工厂更靠近现场,因此中间级维修比起将失效系统送回工厂维修要简便省时。

第三级维修为工厂级。现场级维修与中间级维修不能解决的问题只能由生产工厂来维修。这时失效的系统或部件被送回工厂进行维修。

上述 3 级维修的修复率或平均修复时间通常相差很大。用于计算易维护性的 MTTR 是现场级维修的平均修复时间,它远小于中间级维修的平均修复时间,而中间级维修的平均修复时间又远小于工厂级维修的平均修复时间。

为了寻求提高系统可维护性的途径,有必要进一步分析对失效系统进行修复所需时间的组成。一般可将系统修复时间分为两大部分:被动修复时间和主动修复时间。被动修复时间是指从发现系统存在故障至维修人员开始进行维修的这段时间,其长短取决于管理水平和技术支持能力。主动修复时间是指维修人员实际进行修理所花费的时间,它又可进一步分为下列几部分:①故障检测时间,即用测试手段确认存在故障所需的时间;②故障定位时间,即用测试手段确定故障位置所需的时间;③修理时间,即更换故障部件或修复故障部件所需的时间;④验证时间,即验证系统或部件确实已被修复、能正常工作所需的时间。主动修复时间与维修人员的技术水平有重要关系,但另一方面可通过改进系统的硬件与软件设计,完善故障检测和诊断措施来达到缩短主动修复时间的目的。

为了提高系统的可维护性,一方面应提高管理水平和改善后勤支持,以缩短被动修复时间;另一方面,应在设计系统时就考虑使系统易于测试、诊断和修理,包括采用自动诊断技术、可测试性设计技术、内装自测试技术以及系统结构的模块化技术等,以缩短主动修复时间。此外,计算机系统在发布时备有故障诊断程序和恢复系统等软件对于实现其可维护性也至关重要。

随着计算机系统的正常工作越来越依赖于其软件系统,计算机系统软件的可维护性日益成为计算机可维护性中一项重要的内容,易维护性的概念也得到了扩展。国际标准化组织(International Organization for Standardization,简称 ISO)和国际电工委员会(International Electrotechnical Commission,简称 IEC)成立的信息技术领域的联合委员会(JTC 1)于 2003 年起草的规范技术报告(软件工程产品质量第二部分-外部标准 ISO/IEC TR 9126)对可维护度进行了扩充。在计算机系统的维护过程中,维护者、用户和包含该软件的系统需要消耗一定的工作量或耗费一定的资源都被认为是度量可维护性的因素。在

ISO/IEC TR 9126 中维护性度量包括多种度量：可分析性度量（analysability metrics）、可改变性度量（changeability metrics）、稳定性度量（stability metrics）、可测试性度量（testability metrics）、维护性的依从性度量（maintainability compliance metrics）。ISO/IEC 9126 对这几个方面进行了标准的定义和详细的说明，并还给出了与每项度量相关的测量方法、测量公式及数据元素计算、测量值解释、度量标度类型、测量的类别、测量输入、目标用户等。

参考文献

1. IEEE Standard Computer Dictionary, 610. 12, 1990

2. ISO/IEC TR 9126: Software engineering-product quality. International Organization for Standardization, 2003

3. Johnson B W. Design and analysis of fault-tolerant digital systems. Reading, MA: Addison-Wesley, 1989

（陈汉华　胡谍）

jisuanji xitong keyongxing

计算机系统可用性（availability of computer system）　反映用户在特定环境中有效、高效并且满意地访问计算机系统特定信息或资源的能力（ISO 9241-11 标准）。计算机系统可用性是衡量计算机系统的重要质量指标，不仅涉及界面的设计，也涉及整个系统的技术水平。可用性测试是通过用户操作各种任务去评价的，强调环境因素的重要性，往往包含用户类型、具体任务、操作环境等方面，并且需要考虑非正常操作的情况。

计算机系统可用性用于衡量任意时刻计算机系统是否准备好、是否能正常工作。通常，我们用系统保持正常运行时间的百分比来量化计算机可用性，其一般表达式为：

$$\text{计算机系统可用性} = \frac{\text{系统正常运行时间}}{\text{运行系统总时间}} \times 100\% = \frac{\text{运行系统总时间} - \text{系统宕机时间}}{\text{运行系统总时间}} \times 100\%$$

更具体而言，我们可以通过计算机系统的可靠性和可维护性来衡量计算机系统的可用性。计算机系统的可靠性用平均无故障时间（MTTF）来度量，即计算机系统平均能够正常运行多长时间，才发生一次故障。系统的可靠性越高，平均无故障时间越长。可维护性用**平均修复时间**（MTTR）来度量，即系统发生故障后维修和重新恢复正常运行平均花费的时间。系统的可维护性越好，平均维修时间越短。从而，计算机系统的可用性定义为

$$\text{计算机系统可用性} = \frac{\text{MTTF}}{\text{MTTF} + \text{MTTR}} \times 100\%$$

计算机系统可用性的高低直接决定了计算机系统正常完成任务的可能性大小。根据计算机系统可用水平的高低，计算机产业界通常把计算机系统可用性划分成容错可用性、极高可用性、具有故障自动恢复能力的可用性、高可用性以及商品可用性等五个级别，如下表所示。

可用性级别	可用水平（%）	每年宕机时间
容错可用性	99.9999	小于 1 min
极高可用性	99.999	5 min
具有故障自动恢复能力的可用性	99.99	53 min
高可用性	99.9	8.8 h
商品可用性	99	43.8 h

在工程实现上，可以通过硬件冗余或者软件的方法来提高计算机系统的可用性。硬件冗余指的是在系统中维护多个冗余部件来保证工作部件失效时可以继续使用冗余部件来提供服务；而软件的方法则是通过多系统内的部件或者机器进行状态监测，当发现某个部件或者某台机器失效时启动备用部件或者机器接管失效部件或者失效机器的工作来继续提供服务。无论是硬件还是软件的方法，在设计高可用性的计算机系统时都必须遵循以下 3 条原则：

（1）排除单点的故障，通过增加冗余组件的方式，保证单个组件的故障不会导致整个系统的故障；

（2）保证交叉点的可靠，在多线程系统中，交叉点往往容易成为故障点，具有高可用性的系统必须对交叉点的可靠性提供保障；

（3）系统故障的及时检测，当系统发生故障时，必须能及时发现并采取相关应对手段。

关于计算机系统可用性的国际标准有：

（1）ISO 9241-11（涵盖有效性、效率和用户主观满意度等方面）

（2）ISO DTR 16982（可用性方法支持）

（3）ISO/IEC 9126-1（涵盖功能性、可靠性、可用性、有效性、维护性、移植性等方面）

（4）ISO/AWI 16071（人机交互的人因工效

学——软件可用性指南）

参考文献

1. Gray J, Siewiorek D P. High-availability computer systems. Computer, 1991

2. IBM Global Services. Improving systems availability, IBM Global Services, 1998 （金海）

jisuanji xitong weihu

计算机系统维护（system maintenance of computer） 为了满足用户需求，保障系统正常运行达到既定目标，计算机软、硬件技术人员对系统整体或局部进行的故障错误清理、系统修改与完善等工作。

系统维护面向系统中各个构成因素。按维护对象不同，系统维护的内容可分为以下 4 种类型：

（1）硬件维护 主要指对主机及外设的日常维护和管理，如机器部件的清洗、润滑，设备故障的检修，易损部件的更换等。早期的计算机多采用面板维护方式。在面板上设置若干开关、板键、指示灯等元器件，由手工方式进行维护。随着计算技术的发展，维护技术也有了很大的提高。目前在较大的系统中一般均设有维护处理机支持系统的调试和维护，较小的系统则通过内置维护诊断固件或使用一些专门的自动检查技术进行维护。计算机硬件故障则多采用诊断技术，由计算机自动实现故障检测和定位。诊断方式可以采用自诊断、互诊断或他诊断模式。

（2）环境维护 系统的业务处理是基于运行环境的正常支撑而实现的，一旦运行环境发生问题或变化，必然地引起系统的修改和调整，因此需要定期检查、维护系统的运行环境。

（3）软件维护 按 2006 年国际标准化组织发布的 ISO/IEC 14764 中规定，将软件维护的不同性质划分为正确性维护、适应性维护、完善性维护与预防性维护。诊断和修正系统中遗留的错误，就是正确性维护。适应性维护是为了使系统适应环境的变化而进行的维护工作。在系统的使用过程中，用户往往要求扩充原有系统的功能（如增加一些在软件需求规范书中没有规定的功能与性能特征、对处理效率和编写程序的改进等），为了满足这些要求而进行的维护工作就是完善性维护。系统维护工作不应总是被动地等待用户提出要求后才进行，还应进行主动的预防性维护，即选择那些还有较长使用寿命，目前尚能正常运行，但可能将要发生变化或调整的系统进行维护，目的是通过预防性维护为未来的修改与调整奠定基础，这些工作就是预防性维护。

（4）数据维护 系统业务处理对数据的需求是不断发生变化的，除了系统中主体业务数据的定期正常更新外，还有许多数据需要进行不定期的更新，或随环境或业务的变化而进行调整以及数据内容的增加、数据结构的调整。此外，数据的备份与恢复等，都是数据维护的工作内容。

系统维护工作的特点有：

（1）结构化的开发方法 如果系统开发没有采用结构化分析与设计方法，则相应的维护也只能是非结构化维护。如果系统开发采用了结构化方法，则系统交付时有完整的配置文档、维护系统接口等特点，将大大减少维护的工作量，提高系统质量。

（2）高素质的维护人员 因系统维护所要解决的问题可能来自系统整个开发周期的各个阶段，承担维护工作的人员应对开发阶段的整个过程、每个层次的工作都有所了解，从需求、分析、设计一直到制造、编码、测试等，并且应具有较强的系统调试和排错能力，这些对维护人员的知识结构、素质和专业水平有较高的要求。

（3）不断出现的新问题 系统维护中的绝大部分问题源于系统分析和设计阶段，理解他人的设计很难，而且这种难度随着系统配置文档的减少而增加。同时，系统维护人员的不稳定也会带来很多非技术性的问题。

（4）大量的组织管理 系统维护工作不仅仅是技术性工作，为了保证系统维护工作的质量，需要付出大量的管理工作。系统维护工作，首先必须建立相应的组织，确定进行维护工作所应遵守的原则和规范化的过程；其次应建立一套适用于具体系统维护过程的文档及管理措施以及进行复审的标准。维护计划的内容应包括维护工作的范围、所需资源、确认的需求、维护费用、维修进度安排以及验收标准等。此外，还应注意系统维护的限度问题。系统维护是在原有系统的基础上进行修改、调整和完善，使系统能够不断适应新环境、新需要。但一个系统终会有生命周期结束的时候，当对系统的修改不再奏效，或修改的困难很多且工作量很大、花费过大以及改进、完善的内容远远超出原系统的设计要求时，就应提出研制新系统的要求，从而开始一个新的系统生命周期。

参考文献

1. 李海泉. 电子计算机系统的故障诊断及维

护系统. 北京：机械工业出版社，1984

2. April Alain, Abran A. Software Maintenance Management: Evaluation and Continuous Improvement. New York: Wiley-IEEE Computer Society, 2008

3. Grubb P, Takang A A. Software maintenance: concepts and practice. New Jersey: World Scientific Publishing, 2003 （赵峰　杨肃英）

jisuanji xitong xingjiabi

计算机系统性价比（computer system cost performance ratio） 指单位价格所能获得的计算机系统性能（参见**计算机系统性能评价**）。具体表示为：计算机系统性价比 = 计算机系统性能/计算机系统价格。它体现了性能评价与成本分析的综合结果，可被用来指导新型计算机系统的设计和改进，包括选择计算机类型、型号和确定系统配置等。计算机系统性价比应该建立在对产品性能要求的基础上，也就是说，先满足性能要求，再谈价格是否合适。由于性价比是一个比例关系，它存在其适用范围和特殊性，不能一概而论。

计算机发展初期，人们用简单的技术指标（如吞吐率、存储容量、响应时间等）描述计算机性能，用简单的测量方法收集计算机运行信息。随着计算机系统不断更新，系统性能问题日趋复杂，逐步开展硬件和软件监测工具的研究，应用概率论、排队论建立系统分析模型的研究以及应用数字模拟技术进行计算机系统模拟。引入计算机系统性价比指标能够在兼顾成本与价格基础之上，为生产商和用户提供一种数量化的、能进行度量和评比的客观指标。计算机系统性价比评价是一项重要工作，其最终目的是使计算机系统的设计、制造和使用形成有机的系统工程整体。

参考文献

1. Bryant R E, O'Hallaron D. 深入理解计算机系统. 2 版. 龚奕利，雷迎春，译. 北京：机械工业出版社，2011

2. Hennessy J L, Patterson D A. 计算机体系结构：量化研究方法（原文第 5 版）. 贾洪峰，译. 北京：机械工业出版社，2012 （李克秋）

jisuanji xitong xingneng/gonghaobi

计算机系统性能/功耗比（computer system performance/power ratio） 用来衡量计算机系统单位功耗所能获得的性能（参见**计算机系统性能评价**）。具体表示为

$$\frac{\text{计算机系统性能}}{\text{功耗比}}=\frac{\text{计算机系统性能}}{\text{计算机系统功耗}}$$

现代计算机的功耗会因计算机当时所执行的应用程序或任务的不同，而有很大差异。在任意给定的时间内，实际的功耗主要由硬件功耗和软件功耗两个方面决定。所谓硬件能耗，是指针对特定工作，计算机一段时间内的能量消耗；而硬件功耗为这段时间能量消耗的速率。软件功耗是指由于软件的运行而引起的消耗，其值为系统功耗与硬件功耗之差，取决于计算机所执行的任务和软件（包括所使用的操作系统和应用程序）。在完成一个任务时，比如对系统资源的管理任务、对高级语言的编译任务或对某项具体工程应用的任务，不同的设计思想会产生不同效果的软件。由此，我们可以通过合理硬件搭配以及使用高效的软件系统来提高计算机系统性能/功耗比。

参考文献

Bryant R E, O'Hallaron D. 深入理解计算机系统. 2 版. 龚奕利，雷迎春，译. 北京：机械工业出版社，2011 （李克秋）

jisuanji xitong xingneng moni

计算机系统性能模拟（computer system performance simulation） 指利用软件模拟计算机系统的操作，进而评价其性能的一种技术。模拟程序能够动态地表达计算机系统的运行状态，并基于统计得出系统的性能指标。模拟过程通常将计算机系统描述为一个复杂的排队系统，用离散状态和不同时刻的瞬时状态（事件）来模拟计算机系统的特征，从而获得性能评价所需要的参数，例如利用率、吞吐率、响应时间等。为便于建立模型和进行模型的有效性检验，人们企图使模拟模型在时间和空间上与真正的系统有一定程度的相似关系。在模拟过程中，希望能方便地改变参数甚至改变模型的结构，并能通过键盘命令随时输出数据和图表。因此，计算机模拟要求计算机有很强的并行处理能力，有较高的运算速度，有人机交互能力和便于使用的模拟语言。

根据描述的方式不同，模拟分为面向事件、面向活动和面向进程三种方式。面向事件方式通过按序执行事件程序所引起系统状态的变化来实现。事件依发生时刻的顺序排列，模拟系统依次从事件表中

选择最早发生的事件,激发该事件并将模拟时钟推进到该事件发生的时刻,如此依次执行直到模拟终止。面向活动方式利用活动来表示两个可以区分的事件之间的过程,它标志了系统状态的转移,活动是否发生由规定的事件是否满足激活条件而定。面向进程方式将计算机的系统特性描述成一组异步的、同时发生并且相互作用的进程,进程由事件的时间序列及若干活动组成。面向进程方式接近实际问题但工作机制复杂,而面向事件方式工作简单且易于理解,是现在常用的方法。

系统性能模拟一般按以下 4 个步骤进行,即建立模拟模型、编制模拟程序、执行模拟程序、对模拟结果进行统计分析。上述各方面可能需要反复进行多次,直到得出满意的结果。其中,结果分析是模拟工作的重要环节。从分析结果的观点来看,可分成终止型模拟和稳态型模拟。终止型模拟的运行长度是事先确定的,因为模拟时间有限,系统初始状态直接影响系统的性能,所以要求每次运行的初始条件相同。为了消除初始状态的影响,还必须将模型多次独立地运行,其次数应达到一定要求。稳态型模拟需要有足够长的运行长度,其目的是估计系统的稳态性能。由于运行时间长,初始状态的影响可以忽略。根据对置信区间精度的要求,分别采用固定样本长度法、序贯法、均值法、稳态型序贯法、重复产生法及重复删除法进行分析。

编制模拟程序时可采用各种模拟语言,而模拟语言是一种描述系统模型的高级编程语言,一般是在别的通用编程语言的基础上建立的,提供表示系统基本单元、部件和调度操作的模块。模拟语言可分为离散事件模拟语言(如 GPSS 及其各种改型、SIMSCRIPT、GASD、CSL、SIMULA 等)和连续系统模拟语言(如 DARE、ACSL、CSS、CSSL 等)两大类型。

参考文献

1. Bryant R E, O'Hallaron D. 深入理解计算机系统. 2 版. 龚奕利,雷迎春,译. 北京: 机械工业出版社, 2011

2. Tanenbaum A S. 结构化计算机组成. 4 版. 刘卫东、徐恪,译. 北京: 机械工业出版社, 2001

(马范援　郑衍衡)

jisuanji xitong xingneng pingjia

计算机系统性能评价(computer system performance evaluation)　为了某种目的,选用一定的度量项目,通过实测或通过建立模型对计算机系统的性能进行测试,并对测试结果做出评价的技术。

系统性能指标是反映计算机系统性能和特征的一组参数。常用的性能指标有运算速度、加速比、吞吐率、响应时间、利用率、性能价格比等。其中运算速度通常用每秒执行指令的条数(MIPS)或者每秒运行浮点运算的次数(MFLOPS)来表征(参见**运算速度评价**)。

不同用户对系统性能关心不同的指标,例如科学计算用户最关心 MIPS、MFLOPS、加速比;军事用户最关心可靠性和环境适应性;过程控制人员最关心实时性和可靠性;维护人员最关心可用性和可维护性。

必须注意,上述性能指标均与系统特性和工作负载有关,评价者必须清楚是在什么系统和什么样的负载下测得的性能指标。

计算机性能评价所采用的评价方法大致可分为两类,即测量法和模型法。

测量法通过一定的测量设备或测量程序,测得实际运行的计算机系统的性能指标或有关参数,然后对它们进行统计分析以求出相应的性能指标,这是最直接的性能评价方法。但是这种方法只能适用于已经存在并运行的系统,而且比较费时。

模型法首先对要评价的计算机系统建立一个适当的模型, 然后求出模型的性能指标, 以便对系统进行评价。此法既可用于已有的系统,也可用于尚未存在的系统,可以比较方便地应用于设计和改进,工作量一般比测量法要小。模型法与测量法是相互联系的,在模型中使用的一些参数,往往来源于对实际系统的测量结果。

用户为测试计算机系统性能所需的一组典型程序称为**基准程序**。用户可按不同需求使用不同的基准程序,所测得的各种指标可供用户对系统性能的好坏进行判定和比较。基准程序按照应用类型可分为科学计算、商业应用、网络服务、多媒体应用和信号处理等类型。此外,基准测试程序还可以分为宏基准测试程序和微基准测试程序。前者测试的是计算机系统的总体性能,针对某种应用类型,它可对不同系统加以比较,因而对系统采购人员很有用,但它不能揭示系统运行好坏的原因。微基准测试程序测量一个计算机系统的某一特定方面的性能,如 CPU 速度、存储器速度、I/O 速度、操作系统的性能等。

目前国际上流行的基准程序可以分为以下 5 类：①综合型（如 Dhrystone，Whetstone 等）；②核心型（如 Livermore Fortran Kernels，NASA 的 NAS 等）；③数学库（如 LINPACK，FFT 等）；④应用型（如 SPEC，Perfect，Splash 等）；⑤并行型（如 NAS 的 NPB，PARKBENCH 等）。

因为计算机系统的飞速发展，测试领域的进一步拓宽，基准程序更新加快，用户需要不断调整所使用的基准程序。

参考文献

1. 陈兴业. 计算机系统性能评价. 广州：华南工学院出版社，1987

2. 李三立，李亚民. RISC——单发射与多发射体系结构. 北京：清华大学出版社，1993

3. 罗晓沛，侯炳辉. 系统分析员教程. 北京：清华大学出版社，1992

（吴霭成　郑纬民　陈玉霜　詹文岛）

jisuanji xinpian

计算机芯片（chips for computer）　用作制造计算机的基本元器件的各种半导体集成电路。从计算机分代的角度看（参见**计算机硬件**），第三代计算机采用的中小规模集成电路和第四代计算机采用的大规模、超大规模集成电路都属于计算机芯片的范畴。

集成电路的概念

集成电路是指通过微电子工艺将多个元器件集成在单个半导体基片上并用外壳包封，形成了能执行特定功能的电路。通常所称的集成电路至少包含一个有源元件，即能对电压、电流起控制作用并具有非线性特性的元件，如晶体三极管。

集成电路（integrated circuit，IC）中最重要的元件是用半导体材料制成的各种晶体管。半导体材料有 P 型和 N 型两种，把这两种半导体结合起来就可做成晶体管等半导体器件。目前常用两种类型的晶体管：一类是**双极型晶体管**，如 PNP 晶体管、NPN 晶体管；另一类是**单极型晶体管**，又称**场效应晶体管**（FET）。计算机芯片中使用的场效应晶体管大部分是一种绝缘栅场效应晶体管，又称**金属-氧化物-半导体场效应晶体管**（MOSFET），MOSFET 又分为 PMOS 晶体管和 NMOS 晶体管。双极型晶体管工作速度高，但结构复杂，所以集成度低；而 MOSFET 具有输入阻抗高、噪声小、抗辐射能力强、制造工艺简单的特点，容易实现高集成度，但工作速度较低。

根据所用晶体管的结构与工作原理的不同，集成电路可分为两大类。一类是基于双极型晶体管的，通常采用晶体管-晶体管逻辑（TTL）或射极耦合逻辑（ECL）形式。另一类是基于金属-氧化物-半导体（MOS）工艺的，有 PMOS，NMOS 以及互补金属-氧化物-半导体（CMOS）等。此外，还有一些集成电路将双极电路和 MOS 电路做在同一芯片上，便出现了双极结构型场效应晶体管（BiFET）、双极金属-氧化物-半导体（BiMOS）、双极绝缘栅场效应晶体管（BiGFET）和线性 CMOS（LinCMOS）等电路形式。

集成电路按其不同功能，可分为数字逻辑集成电路、模拟集成电路和数模混合信号集成电路。数字逻辑集成电路是基于布尔代数，以二进制记数为基础进行数字和逻辑运算的一类集成电路，一般由各种门电路和记忆元件组成。模拟 IC 所处理的信息为连续变化的物理量如电压、电流、温度等。数模混合 IC 既包括数字电路又包含模拟电路，因此又可分为两个系列，即混合信号集成电路和数字化模拟 IC。

绝大部分计算机芯片都属于数字集成电路。数字集成电路主要由门电路组成，因此可按每个芯片上集成的等效门电路个数或元器件个数表征该芯片的集成度。通常按集成度将集成电路（IC）分为 SSI、MSI、LSI、VLSI、ULSI、GLSI 等 6 类（参见**逻辑集成电路**）。

计算机芯片的发展历程

集成电路的设计思想是英国的 G. W. A. Dummer 首先提出的。1958 年美国 TI 公司宣布制成第一块集成电路，该电路仅有 12 个元件。从此，电子工业进了 IC 时代，计算机工业也随着集成电路的持续发展，产生了翻天覆地的变化。

与分立的晶体管电路相比，IC 的优点是：①体积小，节省电路板空间；②全部电路元件位于同一基片上，温度和性能良好匹配；③元件间内部连线短，优化了速度和性能特性；④外连线减少，增加了电路可靠性；⑤功耗和热耗散较低。所以，集成电路一出现，就在计算机中得到了广泛的应用。

与普通集成电路一样，随着微电子工艺的不断进步，计算机芯片经历了从中小规模集成电路，到大规模集成电路，再到超大规模集成电路这样一个持续发展的过程。

在中小规模集成电路时代，出于工作速度的考虑，计算机芯片主要采用双极型工艺制造，最常用的是晶体管-晶体管逻辑（TTL）和射极耦合逻辑

(ECL)这两种电路结构。ECL 电路因为速度很快但耗电较大,所以只用在超级计算机和大型计算机中;而 TTL 电路的耗电较少,但速度稍慢,在中小型计算机中用得较多。因为各类计算机和电子设备的大量需求,出现了多种标准化的中小规模集成电路系列,如摩托罗拉的 M10K ECL 电路、仙童的 F100K ECL 电路、得州仪器的 74/54 系列 TTL 电路等。每个系列均包含不同类型的门电路、选择器、触发器、锁存器、先行进位电路等,总数在 100 种左右,以方便设计者选用。

大规模集成电路时代计算机芯片的主要形式是门阵列,辅以中小规模的标准电路。这时,双极型电路虽然还在使用,但其功耗较大、不易大规模集成的缺点逐渐显露。相反,MOS 电路的优点日益明显,而且,随着工艺的改进,其工作速度也有了较大的提高。进入超大规模集成电路时代后,CMOS 成为主流的半导体工艺,双极型基本退出历史舞台。

CMOS 集成电路技术始终是沿着集成度提高、圆片直径增大、特征尺寸减小、互连线层数增多等方向发展。迄今为止所遵循的主要规律是摩尔定律,即芯片的集成度大体上每 18 个月翻一番。实际的情况是每个芯片上的晶体管数每年增加 50%,或每 3.5 年增加 4 倍;特征尺寸(沟道长度)、门延迟、连线的步径(线宽加间距)每年减少 13%。

作为最典型的两种计算机芯片,**微处理器**(MPU)和动态随机存取存储器(DRAM)在 CMOS 集成电路的发展历程中,最具代表性。20 世纪 70 年代初问世的第一代微处理器 4004,采用 10 μm PMOS 工艺,总线位数 4 位,工作电压 12 V,集成度(每块芯片集成的元器件数)为 2300 个,时钟频率 0.108 MHz。经过 6 μm、3 μm、1.5 μm、1.0 μm、0.8 μm、0.5 μm、0.35 μm、0.25 μm、0.18 μm 等阶段,2002 年推出的 Pentium4 采用 0.13 μm,集成度为 55×10^6 个,总线位数 64 位,时钟频率达 3000 MHz,在 31 年的时间里,特征尺寸缩小 77 倍,集成度提高 2.4 万倍,时钟频率提高 2.8 万倍。动态随机存取存储器的发展表现出同样的态势,从 1970 年的 1 kb 发展到 21 世纪初的 512 Mb,容量提高 51.2 万倍,特征尺寸从 8 μm 发展到 0.09 μm,缩小 89 倍,集成度从 4×10^3 发展到 524.288×10^6,提高 13 万倍。表 1 给出了 Intel 公司生产的单核微处理器芯片的发展历程,表 2 是 DRAM 芯片的发展情况。进入 21 世纪后,微处理器进入了多核时代,表 3 列出了一些典型多核处理器的情况。

表 1　Intel 单核微处理器发展进程表

年份	产品型号	线宽/μm	晶体管数(10^3)	管芯面积/mm^2	字长	时钟频率/MHz	电压/V
1971	4004	10	2.3	13.5	4	0.108	12
1972	8008	10	3.5	15.2	8	0.2	12
1974	8080	6	6	20.0	8	2	12
1976	8085	3	6.5	20.0	8	0.37	5
1978	8086	3	29	28.6	16	5~10	5
1979	8088	3	29	28.6	8/16	5~8	5
1982	80286	1.5	134	68.7	16	6~12	5
1985	80386DX	1.5	275	104	32	16~33	5
1989	80486DX	1.0	1 200	163	32	25~50	5
1992	80486DX2	0.8	1 200	81	32	50~66	5
1993	Pentium	0.8	3 100	264	32	60~66	5
1994	80486DX4	0.6	1 600	87	32	75~100	3.3
1995	Pentium Pro	0.35	5 500	310	32	150~200	3.3
1997	Pentium Ⅱ	0.35	7 500	209	32	233~300	2.8
1998	Celeron	0.25	19 000	154	32	300~333	2.0
1999	Pentium Ⅲ	0.18	28 000	140	32	500~733	1.65

续表

年份	产品型号	线宽/μm	晶体管数(10^3)	管芯面积/mm^2	字长	时钟频率/MHz	电压/V
2000	Pentium 4	0.18	42 000	224	32	1400～2000	1.7
2001	Itanium	0.18	25 000	300	64	733～800	1.6
2002	Pentium 4	0.13	55 000	131	32	2000～3000	1.5
2002	Itanium 2	0.13	410 000	374	64	1300～1600	1.3
2003	Pentium 4	0.09	125 000	81	32	2800～3800	1.2

表 2　DRAM 芯片发展进程表

年份	容量/存储位数	线宽/μm	晶体管数(10^3)	管芯面积/mm^2
1970	1 kbit	8.0	4	9.7
1974	4 kbit	8.5	8	14.5
1976	16 kbit	5.0	16	19.4
1979	64 kbit	3.0	66	31
1984	256 kbit	1.5	262	27
1987	1 Mbit	1.0	1 049	50
1991	4 Mbit	0.5	4 194	44
1994	16 Mbit	0.35	16 777	54
1997	64 Mbit	0.25	67 109	75
2000	128 Mbit	0.18	131 072	70
2002	256 Mbit	0.13	262 144	68
2004	512 Mbit	0.09	524 288	59
2007	1 Gbit	0.075	1 048 576	60
2009	2 Gbit	0.065	2 097 152	70
2011	4 Gbit	0.035	4 194 304	69.3

表 3　典型单片多核微处理器

年份	产品型号	工艺线宽	时钟频率/热设计功耗	核数	重要特征
2001	IBM POWER 4	180 nm	1.3 GHz/125W	2	私有 8 路超标量乱序处理器核，L1：I32 KB + D64 KB。共享 1.5 MB L2 和 L3 控制器。1.74 亿晶体管
2005	AMD Opteron Denmark	90 nm	1.6～2.8 GHz/110 W	2	私有 L1：I64 KB + D64 KB，L2：1 MB。共享 DDR 内存控制器；1GHz HT 接口
2005	Intel Itanium Montecito	90 nm	1.4～1.6 GHz/104 W	2	私有 2 路阻塞多线程处理器核，L1：I16 KB + D16 KB，L2：I1 MB + D256 KB；共享 L3：12 MB；系统总线频率 400～533 MHz
2006	Intel Core 2 Duo Conroe	65 nm	1.8～3 GHz/65 W	2	私有 L1：I32 KB + D32 KB；共享 L2：2～4 MB，前端总线速率：800～1333 MT/s
2007	AMD Opteron Barcelona	65 nm	1.7～2.5 GHz/119 W	4	私有 L1：I64 KB + D64 KB，L2：512 KB。共享 L3：2 MB；DDR2 内存控制器；1.6～2 GHz HT3 接口

续表

年份	产品型号	工艺线宽	时钟频率/热设计功耗	核数	重要特征
2008	AMD Phenom X3 Toliman	65 nm	2.1～2.5 GHz/95 W	3	私有 L1：I64 KB＋D64 KB，L2：512 KB。共享 L3：2 MB；双通道 DDR2-1066 内存控制器；1.6～1.8 GHz HT 接口
2008	Intel Core i7 Bloomfield	45nm	2.67～3.2 GHz/130 W	4	私有 L1：I32 KB＋D32 KB，L2；256 KB。共享 L3：8 MB；4.8 GT/s QPI 总线；3 个 DDR3-1066 内存控制器
2010	AMD Opteron Lisbon	45 nm	1.7～2.8 GHz/95 W	6	私有 L1：I64 KB＋D64 KB，L2：512 KB。共享 L3：6 MB；DDR3-1333 内存控制器；2 个 3.2 GHz HT 接口
2010	IBM POWER 7	45 nm	2.4～4.25 GHz/200 W	8	私有 4 路 SMT 处理器核，L1：I32 KB＋D32 KB，L2：256 KB。共享 L3：32 MB；两个 4 通道 DDR3 内存控制器（总带宽 100 GB/s）；SMP 连接带宽 360 GB/s；12 亿晶体管
2011	Intel Xeon Westmere-EX	32 nm	2～2.4 GHz/130 W	10	私有超线程处理器核，L1：I32 KB＋D32 KB；L2：256 KB。共享 L3：24～30 MB；4 个 6.4 GT/s QPI 总线；4 个 DDR3-1333 内存控制器

注：L1/L2/L3 分别表示 1 级、2 级、3 级高速缓冲存储器；I/D 分别表示指令/数据。

计算机芯片的分类

根据集成电路在计算机五大部件（运算器、控制器、存储器、输入设备、输出设备）中的位置，可大致把计算机芯片分为处理芯片、存储芯片、接口芯片三大类。它们在电路结构上各有特点。处理芯片主要完成计算机的运算和控制功能，基本上采用数字电路。存储芯片从外部特性看，也是纯数字的，但其内部利用了电路的很多模拟特性，如存储电荷的电容、放大信号的灵敏放大器等。接口芯片因为涉及模拟量的处理，所以既有数字成分又有模拟成分。

在 20 世纪 80—90 年代，即微处理器发展的早期阶段，因为集成度不够的原因，在处理芯片和存储芯片之间，还有许多独立的芯片，如 DMA 控制芯片、总线控制芯片、总线仲裁芯片、存储管理芯片、浮点协处理芯片，等等。集成度进一步提高后，这些芯片已不再单独存在，它们的功能，都已并入处理芯片中。

因为对处理芯片作了进一步细分的缘故，现在一般提到处理器芯片，指的就是用于普通计算机（包括个人计算机、工作站、服务器等）的通用 CPU 芯片。它具备定点运算、浮点运算、多媒体运算的能力，能支持办公、上网、游戏、科学计算、数据处理等典型应用。其他处理芯片还包括：主要用在嵌入式系统中、运行主控程序、起控制主导作用的**微控制器**芯片，为了快速执行卷积、滤波、傅里叶变换等数字信号处理算法而设计的**数字信号处理器**（DSP）芯片，为了快速处理网络内容而设计的**网络处理器**芯片，主要用于加速 2D 和 3D 图形绘制过程的**图形处理器**芯片，等等。

进入 21 世纪后，处理芯片的发展进入了一个新的时期。依赖高主频取得高性能的单核处理器因为无法将功耗控制在合理的范围，遇到了发展的瓶颈。于是发展出了单个芯片上集成多个处理器核的多核处理器芯片。除表 3 列出的多核通用处理器外，嵌入式处理器、数字信号处理器、网络处理器等都有多核的产品问世。学术界和产业界也在研究单片上集成更多（几十甚至几百）处理器核的芯片，称为众核处理器芯片。

计算机中用到的存储芯片主要分为随机读写和只读两种类型，初期只用于内存（或称主存）。**只读存储器**用于存放不经常改变的机器配置、系统自检和操作系统引导代码，随机读写的存储器是供正常运行时存放程序和数据的。随机存储器芯片里，静态存储器速度较快，但因使用元件较多，集成度较

低、成本较高;动态存储器集成度较高但访问速度稍慢。实际系统中常把二者分别用于层次结构存储器中的高速缓冲存储器和主存储器。只读存储器在发展中经历了掩膜只读存储器、熔丝可编程只读存储器、可擦可编程只读存储器等形式。目前最常用的是电可擦可编程只读存储器和快可擦可编程只读存储器(又称闪存)。此两种类型,虽从传统分类上还归入只读存储器中,但已经在一定程度上具备可读可写的能力,且具有断电后不丢失数据的非挥发性。尤其是闪存,因其大容量集成、读写方便的特点,已更多地用于外存储器。

在计算机中,连接处理器芯片和存储器芯片,并维持和外界联系的,是各类接口芯片。其中,系统控制芯片(又称北桥芯片)集成了高速系统总线控制器和内存控制器,用于连接各个处理芯片和存储芯片。外围控制芯片(又称南桥芯片)集成了外围总线控制器和若干低速外围设备的控制器,并可通过多级外围总线,与多种外围接口芯片相联。

计算机芯片的设计与制造

计算机芯片中包含有大量晶体管及其复杂互连,从逻辑设计、电路设计、器件设计、工艺设计、版图设计到掩膜制作、模拟验证等,有巨大的设计工作量和很复杂的过程。其设计和制造所需的设备虽要很大投资,但一旦开发完成并形成生产线,即可实现规模化工业生产,使生产成本降低,特别是当生产进入稳定期,成品率明显提高时,产品价格将大大降低。

20 世纪 70 年代到 80 年代,集成电路的设计理论没有很大的创新。无论是 60 年代至 80 年代的**计算机辅助设计**(CAD),还是 70 年代至 90 年代的计算机辅助工程(CAE),它们解决的只是将已有的电路转化成版图。而从 80 年代中期开始的**电子设计自动化**(EDA)则是从基于硬件描述语言的系统描述开始,按照一定的规则导出逻辑单元间的连接图,再通过自动布局布线导出用于生产的芯片版图,并在生成过程中解决可测试性和低功耗等问题。90 年代中后期开始逐渐演化生成的片上系统设计方法,是一种以 CPU(或 DSP)为核心、以各种已验证的网际协议(IP)核为基础的集成系统设计方法,强调在虚拟的原型设计环境中验证系统并实现系统集成,同时将设计和测试一体化。

集成电路生产由设计、制造和封装三个独立的工序组成。集成电路设计是将系统逻辑与性能要求等转化为具体的物理版图的过程。集成电路的制造是集成电路生产的核心,包括从晶片制备到制成(中间测试)合格电路芯片的过程(参见**集成电路制造**)。集成电路封装是集成电路生产的后工序,即电路芯片制成后进行装架、压焊、密封、检测和打印等。

设计和制造数量不大的专用集成电路时,价格很高。因此,可以预先制造一些具有不同规模和结构的逻辑阵列,然后根据设计要求断开一些内部连线,方便地制作出满足一定应用要求的集成电路,这就是专用集成电路(ASIC)。

随着半导体加工工艺的不断进步,集成规模越来越大,线条宽度越来越窄,已进入纳米范围。集成电路正朝着集成度更高、速度更快、体积更小、耗电量更少的方向发展,同时不断完善和提高各种特性。

多年来,硅材料的集成电路是集成电路的主流,但新材料正表现出很吸引人的优势。比如砷化镓集成电路(参见**砷化镓集成电路**)、锑化铟集成电路、砷化铟集成电路等,它们有比硅半导体电路更高的速度,特别是砷化镓半导体材料是很有希望的材料。

当物质处于低温时,电阻变为零,出现超导现象。利用超导现象可做成的超导集成电路(参见**超导集成电路**),其速度比硅集成电路快得多,且耗电极低。

参考文献

1. 王阳元. 集成电路工业全书. 北京:电子工业出版社,1993

2. Brey B B. Intel 微处理器(原书第八版). 金惠华,等译. 北京:机械工业出版社,2010

(时万春　唐志敏)

jisuanji xueke jiaocha xinjishu

计算机学科交叉新技术(X-computing technology)　随着计算机与信息科学技术的深入发展和应用普及,计算机科学与技术学科与其他相关学科不断相互渗透,从而产生的一系列交叉学科新领域,可以统称为计算机学科交叉新技术(X-computing technology)或 X-informatics。其中,对计算机科学与技术发展的影响较大的有:**生物计算**(bio-computing)、纳米计算(nano-computing)、**社会计算**(social-computing)、**服务计算**(service-computing)等。其他还有**计算语言学**、媒体计算、计算医学工程、计算健康学、企业计算、**计算机游戏**、航天计算、航空计算、计算化学、计算物理、计算力学、计算生态学、计算地理学等。也有人将 X-computing 称为"新型计算机学科"。新型计算机学科将保持计算机科学核

心基础、所需要的相关科学(如物理、离散数学、认知科学、社会学、经济学、生物学等)、专业构成(如计算机工程、软件工程、信息技术应用等),还有广泛的信息学应用领域;新型计算机学科将带来全新的“计算思维(computational thinking)”。

学科发展的动力来自科学理论与工程技术发展的驱动,也来自应用领域的需求牵引。近几十年来,随着计算机与信息学科的不断应用普及与其他学科的渗透,多学科交叉与融合成为促进计算机学科发展的一大动力。随着多学科的相互交叉与渗透,不断出现新领域、新问题、新方法、新思路,跨学科,跨方向的创新理论与成果层出不穷。以 X-informatics 或 X-computing 为代表的计算机学科交叉新技术的形态也逐渐形成。以生物计算、社会计算和服务计算为例,我们可以考察计算机学科交叉新技术的形成过程。生物计算技术与生物信息学作为生物学与计算机科学的交叉学科诞生于 20 世纪 80 年代末,并已经成为 21 世纪最令人瞩目的新领域。它吸引了大量来自两个学科的科学家和研究人员投身于此,产生了许多创新性的成果,给计算机学科带来了生命科学方面的影响,也帮助人们利用计算机科学技术揭示复杂的生物学奥秘和解决人类的问题。社会计算技术是 20 世纪 90 年代由计算机科学与社会科学相交叉形成的研究领域,也是近几年计算机新技术的研究热点。它使得计算机科学技术进入了社会学的范畴,丰富了计算机科学的社会学知识;人们可以使计算机科学技术与社会科学理论方法相结合,帮助人类认识和研究社会科学,解决人文社会学与艺术方面的相关问题。服务计算形成于 20 世纪 90 年代,跨越了计算机与信息技术、商业管理与商业咨询服务等领域,消除了商业服务与信息支撑技术之间的鸿沟,从商务服务角度促进计算机科学技术的变革,使得计算机科学技术更加适应服务经济和商务的需求。可以看出,由于 X-computing 或 X-Informatics 的诞生和迅猛发展,计算机科学与工程的内涵与外延都发生了前所未有的变革。

正是由于计算机与信息科学及工程的跨学科交叉与扩展的新趋势,世界发达国家和新兴国家的大学十分注意计算机学科与其他相关学科的交叉与融合,并纷纷成立了跨学科研究所或采用的相应组织形式来促进这一学科的交叉与扩展。各大学以不同形式建立学科交叉研究机构有以下几种形式:①资源牵引型　有不少大学集中投资和资源建设了若干跨学科的研究所,通过提供集中的场所、设备、资金,吸引多学科的教授及其团队来此做研究和教学,形成一批标志性成果。②新型学院系　有些大学建立了跨学科的新型信息科学与技术学院系,把多个相关学科的系或实验室组织在一个学院内,通过组织形态支持跨学科交叉融合。③带头人牵引型　有些大学依托强势学科和知名教授组织由少数学科带头人牵头的跨学科研究实验室或研究所。④交叉学科组队型　有些大学组织多学科的教授建立跨学科的研究团队或研究中心。⑤实验室/学科交叉矩阵结构　不少大学采用了实验室与系(学科)矩阵式的跨学科交叉结构,促进多学科的合作研究与教学。⑥强势学科牵引型　有些大学依托其一流的强势学科,在计算机学科的实验室中吸引外学科的教师与学生加盟,形成跨学科研究的环境。这些跨学科的研究所或机构形成了孕育计算机学科交叉领域新学科的温床。

随着信息化浪潮不断席卷全球和向人类社会的深入渗透,计算机学科交叉新技术层出不穷,并在国家经济、国防与社会发展的各个方面发挥着越来越重要的作用。

目前,计算机学科交叉新技术的主要发展趋势有:①交叉融合化　计算机学科交叉新技术无论在理论方法上,还是在技术实现上均带有浓郁的学科交叉融合特色,跨学科的创新方式不断引起新方向、新发现、新理论与新技术出现或形成在学科边缘或交叉点上。②网络化　迅猛发展的互联网技术影响到了几乎所有的计算机学科交叉新技术的最新发展趋势,使得各种学科交叉新技术都采用网络平台作为其最新的技术支撑或实现工具。③智能化　计算机科学和**人工智能**理论方法成为许多学科交叉新技术得以借力并在交叉领域不断提升其信息处理能力的努力方向。④泛在化　由于计算机无处不在的计算能力,使得各种学科交叉新技术不同程度地向普适泛在的方向发展。⑤应用领域化　应用需求牵引是计算机学科交叉新技术发展的重要特征;如何深入运用计算机科学技术更好地解决相关交叉学科领域的应用问题将是计算机学科交叉新技术发展的永恒的主题。

参考文献

1. Marc Snic. Computer & Information Science & Engineering-What's All This? The Second US-China Computer Science Leadership Summit, Washington DC, 11th July, 2008

2. 徐晓飞. 美国大学计算机与信息科学的跨

学科发展. 中国计算机学会通讯,2009,(2)

（徐晓飞）

jisuanji yiyongxing

计算机易用性(usability of computer) 用户对应用计算机的主观感受和交互性的客观度量。计算机的设计一般需要考虑计算机用户的使用习惯,以满足用户方便、高效地使用计算机的需求。计算机的易用性是决定计算机用户是否愿意使用计算机的重要指标,计算机的易用性包括主观易用性和客观易用性两个方面。可以通过选择一些有代表性的用户,在特定的环境下,执行一组指定的任务,看系统是否容易使用和高效使用来衡量计算机的易用性。给定一组可用性度量的属性,可以选取每个属性度量的平均值,检查这些平均值是否比设定的最低标准要好,同时考虑易用性度量属性的整体分布情况,以区分用户的差异性。例如,用户对计算机的主观满意度的标准范围设定从 1 级到 5 级,如果主观满意度平均值是 4 级,并且至少 50% 的用户应该给出最高评级——5 级,不超过 5% 的用户给最低评级——1 级,那么认为计算机系统在整体上符合易用性的标准。

计算机易用性涉及计算机系统与人交互的各个方面,涵盖了软件系统设计方法学、人体工程学和认知心理学等相关学科,也包括系统安装和维护过程。

从人体工程学标准角度,计算机硬件和环境的易用性,指特定的用户可以使用计算机高效地完成特定的目标的程度。易用性的属性包括有效性、效率和满意度,其中有效性是指用户完成特定目标的精确性和完整性,效率与用户完成特定目标相关的时间、人力和成本等资源相关,满意度是指用户能够舒适和满意地使用计算机。

计算机软件易用性是人机接口(Human-Computer Interaction,HCI)领域的主要技术问题之一。从目标和可操作性标准的角度,计算机软件易用性可以通过定义效率、学习容易度、灵活性和用户主观态度等操作性标准进行度量。计算机软件易用性表达了软件和应用领域的关系,是软件质量的内在要求。计算机易用性在软件设计领域,被计算机软件质量术语替代。

一般来说,计算机易用性包含了容易学习、使用高效、方便记忆、故障少、主观满意度高等五个相关的属性。

(1) 容易学习　是易用性的最根本的属性。尽管有些系统提供培训课程,以帮助新用户熟悉接口功能,但是,容易学习还是系统的基本需求。毕竟大多数用户对一个新系统的第一经验是学习使用它,容易学习的系统能够让用户使用系统快速开始工作。为了衡量初始学习的容易程度,可以选取一些没有使用过系统的新用户,通过成功完成指定的任务的时间来表示操作的熟练程度,记录这些新用户达到特定使用熟练程度的时间。

(2) 高效使用　也是系统易用性的一个属性。能够高效使用系统的有经验的用户具有很强的生产能力,其中,使用效率可以通过专业用户在学习曲线平坦时的稳定操作水平来衡量。

(3) 方便记忆　是对普通用户和由于一些原因临时暂停使用系统的用户很重要的一个接口属性。普通用户间隔一段时间使用系统,不必每次从头学习,只需要基于以前的学习经验记忆如何使用系统。一种衡量系统是否方便记忆的方法是选择指定时间段内没有使用系统的普通用户,针对他们执行标准的用户测试,记录执行一些典型测试任务的完成时间。也可以对用户进行记忆测试,要求用户解释不同命令的功能或者完成某些任务的命令名称。

(4) 低故障率　用户在使用系统的过程中,系统应该很少出现故障,并且如果系统出现故障,应该可以快速恢复。此外,不允许发生灾难性的故障。系统的故障率可以通过计算用户在执行某些特定任务过程中故障次数来衡量。有些故障可以被立即纠正,除了暂时延缓用户执行的任务之外没有其他影响;另外一些故障是灾难性的,用户不能发现这些故障,导致任务执行出错,或者破坏用户的工作,并且难以恢复。

(5) 高满意度　使用系统应该是一件令人愉快的事情,用户使用系统时主观上是满意的,并且喜欢使用该系统。对用户而言,系统除了能够快速完成某件任务,也应该具有一定的娱乐价值。可以通过简单的询问用户的主观意见,或者选取用户能够正确解释的问题进行问卷调查来衡量主观满意度。

计算机易用性的国际、国内标准

关于计算机易用性的国际标准可以分为以下几类:

(1) 产品使用标准(涵盖有效性、效率和满意度等方面,ISO/IEC TR 9126-4,ISO WD 20282);

(2) 产品质量标准(涵盖用户接口和交互,ISO 9241,ISO/IEC 9126, ISO/IEC 11581);

(3) 开发过程质量标准(涵盖系统易用性设计

开发过程，ISO 13407，ISO DTR 16982，ISO /IEC 14598）；

（4）组织易用性设计能力的评价标准（涵盖易用性成熟度模型和交互设计生命周期，ISO TR 18529）。

目前，国内也制定了一些有关计算机易用性的规范，主要涉及人体工程学硬件设计方面，计算机软件易用性方面等。

参考文献

1. Nielsen J. Usability engineering. Boston，MA：Academic Press，1993

2. Shackel B. Human factor for informatics usability. Cambridge University Press，1991

3. Shneiderman B. Designing the user interface：strategies for effective human-computer interaction. Addison-Wesley，1987

（蒋文斌　刘超）

jisuanji yinyue

计算机音乐（computer music）　利用计算机进行音乐信息处理的技术。计算机具有很强的信息处理能力，而音乐虽然最终表现为声波的振动，但声波只是音乐信息的载体，音乐信息本身则完全可以用计算机来处理。

计算机音乐的历史始于1956年。当时美国伊利诺伊大学的作曲家L. Hiller和数学家L. Issacson在ILLIAC Ⅰ计算机的辅助下创作了Illiac弦乐四重奏组曲，首次作出了真正的“计算机音乐”。此后，计算机在音乐领域的应用范围不断拓展，水平也日趋提高。现在，已有很多商品化的计算机音乐系统上市。

音乐信息在计算机内的表示具有不同的层次，粗略地可分为信号层、内部表示层和人-机界面层。信号层上的音乐信息是符合一定标准（例如电子乐器数字化接口（MIDI）标准）的数字指令，可以被配有MIDI接口的任何电子乐器接受并实时转换成具有听觉效果的声音；内部表示层上的音乐信息按某种计算机可读的格式储存并接受处理；人机界面层上的音乐信息则表现为人类可读的乐谱。

计算机音乐的内容是多方面的。一些具有代表性的分支为：

（1）乐谱识别　把印在纸上的乐谱通过光电扫描仪转换成数字形式并对之进行**模式识别**，最后翻译成内部表示，以备将来使用。

（2）乐谱生成（显示或打印）　把音乐的内部表示翻译成人类可读的乐谱，将其在屏幕上显示出来或付诸打印。其中，屏幕显示可以是交互式的，就是说，计算机可以为作曲家提供一个具有“所见即所得”效果的乐谱编辑器。

（3）计算机记谱　把音乐的物理实现记录下来并转换成数字形式，然后根据音色及乐理知识将其分解成多个声部，经分析后形成相应的内部表示，最终转换成人类可读的乐谱。

（4）计算机作曲　①计算机自动生成音乐动机，并发展旋律；②给计算机输入一个音乐动机，由计算机发展旋律；③给计算机输入一个单声部旋律，由计算机编配多声部和声等。要做到这些，必须把相应的旋律学、和声学、曲式学知识变成计算机可执行的程序。

（5）计算机音乐合成　由计算机、MIDI键盘、MIDI合成器、音响等装置可组成方便的音乐创作平台。用户可以在终端以“所见即所得”的方式编辑乐谱，并可以随时听到由合成器模拟的各种音色的乐器按乐谱的要求“演奏”该乐谱的整体音乐效果，对不满意的地方随时进行修改。

（6）计算机音乐作品分析　通过对特定音程、节奏、和声以及曲式结构等的分析，确定音乐作品的风格，包括音乐作品的时代、民族、地域、流派及特定作曲家作品的内在一致性和风格变迁等。计算机可为这方面的研究辅助处理大量信息并提供定量的分析数据。

（7）计算机音乐信息检索　在网络和多媒体技术的支持下，现在计算机音乐信息检索不仅能提供标题、作者、摘要、乐谱和其他背景材料的检索服务，还能通过网络实现远程点播服务。在一台具有多媒体播放功能的国际联网计算机上，可以点播远在千万里之外的音乐文档中的任何曲目；点播输入可以是描述音乐的文本，也可以是用户哼唱输入的音乐，还可以是一小段音乐片段。

计算机音乐随着网络技术、**多媒体技术**和**人工智能**技术的进步而不断发展。有了计算机的帮助，音乐家们可以更方便地从事音乐活动，进入更高的艺术境界。而广大的音乐爱好者和欣赏者以及音乐使用者则可以有更多的机会实践自己的音乐理想，欣赏自己所钟爱的音乐作品。

参考文献

1. Howe H. Electronic music synthesis. New York：W. W. Norton，1975

2. Collins N. Introduction to computer music，

John Wiley & Sons Ltd, 2009 (白硕 许洁萍)

jisuanji yingyong jishu

计算机应用技术(technologies for computer applications) 计算机在各行各业和各种社会活动中的应用所涉及的基本原理、共性技术与有效方法。在计算机科学技术发展的初期,计算机的处理对象主要是数值数据,应用领域较窄。随着计算机科学技术的发展,其应用范围不断扩展,计算机科学技术在系统(硬件和系统软件)及应用两方面有所分工,各有侧重,从而使计算机应用技术有着相对独立的发展,成为计算机科学技术中的一个二级学科。

计算机应用的过程中,通常需要对信息进行采集和管理,建立计算模型并对信息进行加工,再以适当的形式表示处理的结果,最终再对社会活动和生产过程进行指导甚至直接进行控制。计算机应用技术涉及上述信息处理过程的全部内容,虽然范围十分广泛,但其共性技术是对信息的处理和管理。这里所说的信息处理,包括对信息的获取、表示、存储、检索、运算、转换与展现等诸方面的内容。

计算机所处理的信息,一类是数值信息。处理数值信息往往需要进行大量的计算,数值分析、**最优化方法**、**计算几何**等是其支持技术,有关的理论、方法已成为数学学科中的一个重要分支,即计算数学(参见**数值计算**),一般不再列入计算机应用技术学科。另一类是非数值信息,包括文字、声音、图形、图像和影视(活动图像)等。处理非数值信息所涉及的问题大多表示为符号和规则,其支持技术包括语言文字处理、**计算机图形学**和图像处理、**多媒体技术**和**虚拟现实**、**数据库管理系统**、**人工智能**等。

语言和文字是信息、知识和文化的主要载体,它们的处理技术直接影响到计算机在各个领域的应用以及社会信息化的进程。语言文字处理技术是利用计算机对语言文字的音、形、义及其所承载的信息进行处理的理论、算法和技术。其内容包括语言及文字的自动识别与输入、文档表格的编辑排版与生成、言语的计算机合成以及**自然语言处理**(如**机器翻译**、自动摘要、问题回答等)。而中文信息的处理还包括汉字的编码及输入、**中文信息检索**、中文文本自动处理等一些特殊问题。

图形和图像是信息的另一类载体。它们是计算机图形学和**数字图像处理**的研究对象。计算机图形学研究如何在计算机内建立真实景物或虚拟景物的模型,并将这些模型转化为可视图像在图形设备上进行展现。而数字图像处理则研究如何对图像进行增强、复原、变换、重建等处理,或从中提取关于该图像内容的特征信息。虽然这两门学科属于两个不同的领域,但近十多年来由于影视特技、可视分析、增强现实等的迅速发展,计算机图形学和图像处理、计算机视觉的结合日益紧密,它们相互渗透,进一步促进了相关领域的发展。

多媒体计算技术是对文本、声音、图形、图像及影视进行综合处理,使它们建立有机联系并集成为交互型信息系统的一门技术,其中声音和影视数字信号以及它们与其他传统信息载体的综合处理与应用是多媒体技术的核心。

对现实世界景物及其活动进行模拟仿真,是人们进行科学实验、规划设计、培训演练的重要手段。虚拟现实技术的目标就是以计算机技术为核心,结合相关科学技术,生成与一定范围真实世界在视、听、触感等方面高度近似的数字化环境,用户可以借助必要的装备与数字化环境中的对象进行交互作用,相互影响,产生亲临相应真实环境的感受和体验。

信息管理技术主要研究在计算机中如何有效地组织数据以及如何高效、可靠地储存和检索数据。计算机发展初期采用的是文件系统,随着数据处理应用的发展出现了数据库技术。**数据库**是可持久储存在计算机系统内的有组织、可共享的大量数据的集合,管理这些量大、持久和共享数据的核心软件是**数据库管理系统**(DBMS)。

以信息处理和信息管理技术为核心的计算机应用系统,按照它们所提供的服务,可以分成以下几种类型,它们各自具有不同的特性。

计算机信息系统 许多情况下计算机信息系统是一个很笼统的概念,这里特指主要任务是对数据进行采集、储存和处理并以人机交互方式为用户提供信息服务的系统。通常它处理的数据量很大,且绝大部分数据是持久的,可以为多种应用所共享。除了具有数据管理的基本功能外,它还能向用户提供信息检索、统计、事务处理、规划、决策等信息服务。应用最普遍的两种信息系统是事务处理系统和信息储存与检索系统,前者用来处理预先定义的事务并输出相应的处理结果,如订票系统、金融信息系统等;后者用于存储文档类数据,并可根据查询要求向用户提供相应的文档全文或摘要,如科技文献检索系统、联机医学文献分析和检索系统等。目前计算机信息系统正朝系统集成化、结构分布化、信息多

元化、功能智能化、服务公用化的方向发展。

计算机辅助系统　借助计算机在设计、生产、教学等过程中进行有效的辅助性工作，形成一个以人为主导的人机结合的系统，以充分发挥人的创造力，提高效率，降低成本。目前计算机辅助技术已广泛应用于设计、制造、工程以及教育等领域，形成多种不同的计算机辅助系统。虽然计算机辅助技术与各应用领域密切相关，但其主要的辅助功能都是通过以图形为主的人机交互方式来实现的，因此它的发展与计算机图形学有着密切关系。多年来，计算机辅助技术还针对工程数据的管理、交换、规范以及与其他信息系统的互连、互通、互操作等进行研究，使这门技术朝系统化、集成化的方向发展。

计算机仿真系统　**计算机仿真**是利用计算机建立、校验、运行实际系统的模型以得到其行为特性，从而分析研究该实际系统的方法和技术。这里的"系统"是广义的。它包括工程系统，如电气系统、热力系统、计算机系统等，也包括非工程系统，如交通管理系统、生态系统、经济系统等。随着系统的规模日益庞大，结构日益复杂，仿真技术不但能提高效率，缩短研究开发周期，减少训练时间，不受环境及气候限制，而且对保证系统安全、节约开支、提高质量尤具有突出的功效。

计算机控制系统　计算机控制系统是通过不断采集被控对象的各种状态信息，由计算机按照被控对象的模型和一定的控制策略实时地计算和处理后，作为控制信息去推动执行机构，使被控对象自动地、精确地按照预定的规律运行，以减少人工操作，提高生产效率和产品质量。对于大型复杂系统，还可以借助**计算机网络**的支持，实施既有分散，又有集中的计算机分层控制。

嵌入式系统是计算机控制系统的一种类型，它特指将计算机嵌入一个物理系统中，用来感知并控制或维护系统中其他组成部分的性质或关系，以实现总的系统目标。目前已广泛用于飞机、汽车、家用电器、武器装备、通信设备、医疗设备、玩具等。嵌入式系统通常应满足实时性、对外部事件的响应能力、对机械装置或物理过程的控制能力以及高可靠性、低功耗等要求，这就给嵌入式系统的硬件和软件开发带来了新的挑战。

消息交换系统　计算机网络特别是**互联网**的发展，促使一批新的基于计算机网络的通信应用应运而生。它们统称为消息交换系统。例如**电子邮件**、IP 电话、计算机会议、计算机支持的协同工作、网上聊天、微博、远程教学等。以计算机网络为基础的通信相比传统的以模拟技术为主的通信有许多优点。它支持多媒体通信，采用同步与异步方式相结合、点对点通信与多播（广播）通信相结合，可实现智能化的管理与监控等，因而应用形式多样，发展非常迅速。

在各种计算机应用中，系统的友善性和智能化是计算机应用技术追求的两个共同目标。**人机交互技术**和**人工智能**是与此紧密相关的两项关键技术。人机交互技术是依据对人与人、人与系统之间的交互行为的理解，使计算机能够以友善、自然和直观的方式与用户进行信息交换或提供服务的技术。人机交互技术的形成是多学科综合作用的结果。除了计算机科学技术以外，人性因素、人类工程学、工业工程、认知心理学、社会心理学等学科的发展，也对人机交互技术的发展起着重要的作用。人工智能是研究解释和模拟人类智能、智能行为及其规律的一门学科，其主要任务是建立智能信息处理理论，进而设计可以展现人类某些智能行为的计算机系统。

随着计算机与信息科学技术的深入发展和广泛应用，计算机科学与技术学科与其他相关学科不断相互渗透，从而产生了一系列交叉学科新领域。例如，**生物计算**（bio-computing）、**社会计算**（social-computing）、**服务计算**（service-computing）等。这些新的学科领域以计算机科学技术为基础，与相关学科（如生物学、社会学、经济学等）紧密结合，将带来全新的"计算思维（computational thinking）"。

当前，计算机应用已经渗透到人类活动的各个领域，计算机应用系统也由最初的单机系统向网络分布式系统发展。进入 21 世纪以来，随着计算机及相关技术的进一步发展，通信和计算机设备体积越来越小，价格越来越便宜，各种新型传感器也不断出现。同时，由于人们对生产效率、生活质量的不懈追求，人们开始希望能随时、随地、随意（无困难）地享用计算机能力和信息服务，计算机应用正在步入一个新时代——普适计算时代。各种新的计算机应用需求，将进一步推动计算机应用技术的发展，而计算机应用技术的发展也将大大促进计算机更加广泛和深入地应用，从而加快人类社会信息化的进程。

参考文献

1. 授予博士、硕士学位和培养研究生的学科、专业目录. 2005.12. 中国学位与研究生教育信息网（http://www.cdgdc.edu.cn/xwyyjsjyxx/sy/glmd/264462.shtml）

2. Ralston A, Reilly E D, Hemmendinger D. Encyclopedia of Computer Science. 4th ed. John Wiley & Sons, 2003
（张福炎　等）

jisuanji yingjian

计算机硬件（computer hardware）　组成计算机系统的所有物质实体（如元器件、部件、设备、设施），以及设计、制造、检测、维护这类物质实体时涉及的理论、原则、方法、技术所构成的学科。

在计算机问世初期，“计算机”一词实际上只是指“计算机硬件”。进入20世纪60年代，由于程序设计技术的进步，才形成“计算机硬件”和“**计算机软件**”的概念。

发展历程

由于计算机硬件是计算机的物质体现，所以自从计算机问世以来，其发展都是以硬件的变革作为分代的主要标志。它共分为4代。

第一代计算机以前的计算工具是机械式或机电式的。典型的代表是算盘、手摇计算机和机电式计算机。它们的运算速度慢，精确度差，特别是不能自动进行运算，所以不是现代意义上的计算机。

第一代计算机（从20世纪40年代中期到50年代末期）　是电子管计算机。人们普遍把1946年2月开始运行的ENIAC作为第一台计算机的代表。第一代计算机的运算控制部件使用了大量的电子管，不仅运算速度大大提高，而且能实现自动计算。其存储设备最初使用阴极射线管或超声延迟线，以后使用磁鼓存储器和磁心存储器。输入和输出设备则沿袭了当时的机电式高级分类统计装备，如穿孔卡片机、穿孔纸带机和击打式打印机。

由于硬件设备制造成本昂贵，程序长度和数据精度都很有限，计算机的类型和数量也不多，主要用于解决当时较为复杂的科学和工程计算，并且集中于军事方面。

第一代计算机硬件的设计原则是尽量少用电子管。组装工艺采用焊线技术，组装密度低，体积庞大。检测工具主要是电工仪表和示波器。计算机的可靠性差，抗干扰能力低，对机房环境要求高。整机平均故障间隔时间往往不到1小时。

第二代计算机（从20世纪50年代中后期到60年代中期）　是晶体管计算机。1947年晶体管问世，50年代制成晶体管电路。这种电路是第二代计算机的运算和控制部分的主要硬件。与电子管电路比较，晶体管电路具有工作速度快、可靠性高、耗电量少、体积小、成本低廉、适宜大批量生产等突出优点。第二代计算机的存储器曾经使用过多种存储媒体，包括磁膜、磁泡、磁杆等，但它们都只是昙花一现，最后还是稳定在磁心上。第二代计算机的主存储器是磁心存储器。磁鼓存储器和晚些时候出现的磁盘存储器用来作为大容量辅助存储器。它扩大了磁心存储器的存储容量。输入和输出设备增加了磁带机和显示器等。

由于晶体管电路的诸多优点，计算机的结构得到较大发展，出现了多种类型的计算机。虽然它们的数量还不是很多，但却开拓了计算机应用的新领域。由过去以军事为主的应用转入以经济为主的应用领域，由以科学与工程计算为主的应用转入科学与工程计算、事务处理和过程控制等更为广阔的应用领域。

第二代计算机硬件的主要设计原则是提高整机性能，即采用较复杂的逻辑结构，以满足应用对计算机性能和功能的多种要求。计算机制造时采用印制板和绕接连线，从而提高了组装密度，缩小了体积。这就是通常所说的“快（提高运算速度）、大（扩大存储容量）、小（缩小整机体积）”三原则。体积缩小后，散热问题成为工程实施方面研究的重大课题。检测手段采用自检电路（如奇偶检验等），改善了维护条件。抗干扰能力和可靠性也有了提高，平均故障间隔时间可达1000小时以上。

第三代计算机（从20世纪60年代中期到70年代初期）　是集成电路计算机。1958年半导体集成电路问世，1962年有集成电路商品，很快就用于计算机的运算器和控制器。当时的集成电路只能把100个以下的元器件集成在一个芯片上。与第二代计算机相比，第三代计算机大大提高了运算速度，缩小了体积，提高了系统的可靠性。第三代计算机的主存储器仍采用磁心存储器，而作为辅助存储器的磁鼓存储器则逐渐被淘汰，磁盘存储器占据了绝对优势的地位。输入和输出设备变化很大。传统的机电式设备，除击打式打印机外，基本上都被淘汰。磁带机成为重要的输入输出设备。同时，还陆续出现了多种非击打式印刷设备。软磁盘开始被采用作为输入输出设备。由于数据通信的发展，计算机进入网络环境。因此，由电传打字机发展起来的终端设备也成为计算机重要的输入和输出设备。

第三代计算机硬件的主要设计原则是规范化。集成电路的生产宜于数量多而品种少，所以计算机的生产便于实现规范化和自动化。这时，多层印制

电路板和高密度组装已普遍用于制造工艺。自检电路的广泛使用不仅促使较为复杂的检测设备(如逻辑分析仪)的出现,而且由计算机自行检测硬件故障的检测程序和诊断程序也日臻完善。计算机对环境的要求不断降低,可靠性指标大幅度提高,以至于用户逐渐淡薄了平均故障间隔时间这一类概念。

第四代计算机(20世纪70年代中期以来) 是大规模和超大规模集成电路计算机。与第三代计算机比较,第四代计算机所用的集成电路芯片在集成度方面虽然只是量的变化(一个芯片上能集成的元器件在10 000个以上),但在性能方面却产生了质的飞跃。首先,由一个或几个芯片组成的**微处理器**,使得计算机的新成员,即**微型计算机**,进入了人类的社会生活,从而再一次大规模地开拓了计算机应用的新领域。半导体集成电路技术的发展遵循摩尔定律,其集成度每18个月翻一番。如此快的发展成为微处理器性能突飞猛进的主要动力。其次,半导体存储器取代了延续多年的磁心存储器,并解决了将部分程序固化的问题。半导体存储器的容量越来越大,存储密度按每年60%的速度增长。**辅助存储器**仍以磁盘存储器为主,但发展十分迅速,存储密度按每年60%的速度增长。第三代计算机时期发展起来的各种输入和输出设备继续得到使用和发展。终端设备成为人与计算机交互通信的主要手段。图形、图像和声音的输入和输出设备在技术上逐渐成熟,陆续被采用。

第四代计算机硬件的设计已深入到集成电路芯片之内,设计原则仍然在简化逻辑、实现高性能和保持规范化三者之间求取最佳平衡。制造工艺还是采用多层印制电路板和高密度组装技术,但重点已转到集成电路芯片的制造上了。硬件可靠性进一步提高,检测工作主要由测试软件自动完成。随着计算机一代代的发展,它的设计思想、制造工艺和检测技术在不断地进步,标准化工作也在不断地深入。

综上所述,微电子技术、磁记录技术、光电子技术、精密机械技术、高密度组装加工技术等是促进现代计算机发展的关键,而计算机的发展又对这些技术提出更高的要求。计算机硬件的水平通常反映出同一时期电子工业和精密机械工业的水平。

基本内容

计算机硬件包括所用的**计算机芯片**、**计算机逻辑部件**、**计算机存储设备及网络存储**、**计算机输入输出设备**、**计算机工程设计和制造**、**计算机硬件可靠性**、**计算机维护和机房设施**等。现简述如下。

计算机芯片指主要用于制造计算机的各类集成电路,主要是数字集成电路。计算机的各个部件和各种设备都需要使用集成电路,特别是**中央处理器**和存储设备。**微处理器**可将中央处理器集成在一个芯片上,大量存储单元及其外围读写电路也可以集成在一个芯片上。**半导体存储器芯片**分为**随机存取存储器芯片**和**只读存储器芯片**。由于微电子工艺技术的飞速发展,已可将具有完整功能的系统(包含处理器、局部存储器、接口和专用处理部件等)集成在一个芯片上,称为**片上系统**。还可以将几个甚至几十个中央处理器集成在一个芯片上,称为**多核和众核处理器**。另外,还可根据用户的需求设计制作专用芯片。多年来,基于硅器件的半导体集成电路是最广泛使用的。**砷化镓集成电路**、锗硅异质结器件、**光电集成电路**、**超导集成电路**等也处于发展之中。因为半导体芯片在计算机中的大量使用,芯片设计方法也成为计算机硬件设计的一个重要部分。

计算机逻辑部件包括运算器和控制器两部分。运算器是完成各类算术运算和逻辑运算的部件。它通常由能完成加法、乘法、除法、布尔代数运算、移位操作等的电路和存放操作数及运算结果的寄存器组成。在高性能计算机中,还设有能完成浮点运算(加、减、乘、除、乘加、开方等)的**浮点运算器**及能完成定点及浮点向量运算的**向量运算部件**。控制器按照程序规定的顺序,自动接受和执行指令。它解释指令的操作码和生成地址码,并根据译码结果将适当的控制信号送到计算机的有关部件。控制器由**程序计数器**、**指令寄存器**、时序控制部件和组合逻辑电路组成,这种控制器称为**硬连线控制器**。还有一种控制器是通过执行若干条比指令低一层次的微指令所组成的微程序而完成指令的各种基本操作的,这种控制器称为**微程序控制器**。

计算机存储设备用来储存程序、数据、运算结果和资料等。它可以接受数据和保存数据,并根据命令提供数据。根据功能、结构和工作原理的不同,存储设备可分为**半导体存储器**、**磁存储器**、**光存储器**、**外存储子系统**等。半导体存储器主要用来作为计算机的主存储器,即存放计算机运行时随时需要使用的程序和数据。半导体存储器件也可以做成具有磁盘功能的半导体盘。磁存储器是以硬磁材料为存储媒体的一类辅助存储器,可分为**硬磁盘存储器**、**软磁盘存储器**和**磁带存储器**。**光存储器**也是计算机的一种辅助存储器,它主要是利用半导体激光器而做成的光盘存储器。磁带存储器、软磁盘存储器和光存

储器的存储媒体可以脱机保存，易于更换，因此，除了可作辅助存储器外，还可用作输入和输出设备。由于科学技术的发展，需要对数量巨大的数据和资料进行保存和读出，因此出现了外存储子系统。它是由大容量存储设备和控制设备组成的、用以组织和管理数据的存取和传输的辅助存储系统，既可采用传统的外设总线相连，也可以用多种网络与主机相连，使其具有更好的可扩展性。

网络存储指将分布在网络上的存储设备或存储子系统通过某种连接方式实现与服务器或服务器群的互连，提高数据的共享性、可用性、可扩展性和可管理性的数据存取技术。常用的连接方式有三种：**直连存储**、**附网存储**和**存储区域网**。直连存储指直接通过电缆将存储设备与服务器相连的存储方式，也称直接附属存储。系统存取访问的I/O请求直接在服务器和存储设备间进行，存储设备由主机独享。附网存储是一种将存储设备直接连网并使用标准协议提供文件级数据访问的存储方式，其特点就是在物理连接上将存储器直接连接到网络上，避免了服务器不必要的I/O负载。存储区域网是一种连接外接存储设备和服务器的架构。它通过可伸缩的网络拓扑结构互连不同类型的存储设备与服务器，提供内部任意结点间的多路可选择的数据交换，并将存储管理集中在相对独立的区域内，实现最大限度的数据共享和优化管理，以及系统的无缝扩充。目前，直连存储已逐渐被附网存储或存储区域网所替代。网络存储通过**存储虚拟化技术**，实现了存储与计算相分离的原则。当应用需求逐步转向面向数据应用时，网络存储可提供面向数据的应用系统对数据集中、密集数据存取、海量数据、实时数据分发、数据整合、数据管理、数据交换、数据迁移、数据重用、数据安全性、数据可管理性的要求，甚至提供跨平台的信息存储与共享。

计算机输入输出设备把数据输入计算机或把计算机中需要输出的数据传送给用户或其他接收设备。输入输出设备种类繁多，功能广泛。可更换的存储设备，如磁带、软磁盘和光盘等辅助存储器也常用作输入输出设备，它们既可作为输入设备，又可作为输出设备。常用的输入设备有**键盘**、**鼠标**，还有光笔、**触屏**、数字化仪、光学字符阅读机、**条码阅读器**、磁卡机、**扫描仪**、**数字摄像头**、**射频识别阅读器**、**全球导航卫星系统接收器**等。基于触屏的多点触控技术，因可输入手势，更具人性化特点，已被许多移动终端、平板计算机和笔记本计算机采用。常用的输出设备有**显示器**、**击打式打印机**、**非击打式印刷机**、**绘图机**、**投影仪**等。印刷机最初只输出字符，后来能输出图形和图像，可以输出清晰度高和质量极佳的彩色硬拷贝。汉字可以作为一种特殊图形输出，已成为打印设备普遍具备的功能。通过通信线路或数据传输线路和计算机相连的输入输出设备称为**终端设备**，它不但具有人机交互功能，还可使用户在远离计算机的地方和计算机进行人机交互操作。以智能手机为代表的**移动终端**、各种**便携式媒体播放器**、**电纸书**等，已成为人们获取信息的重要手段。

计算机电源是为计算机各个部件正常运行提供能源支持的部件。计算机对电源的尺寸、重量和可靠性等指标要求不断提高。要求交流电源能高质量地稳压稳频，要求**直流电源**不仅抗干扰性能强，而且在负载大幅度变化时仍能保持稳定的电压。计算机系统要求不间断地供电，因此需要**不间断电源**，当交流输入电源的变化超出规定范围或当外界电网紧急断电时，它能够及时接通其他电源，或延续供电一段时间，使计算机中的信息不致遗失或遭到破坏。

计算机的工程设计和制造包括**可靠性设计**、**热设计**、**印制板设计**和制造、**集成电路制造**、**高速数字信号传输**技术、焊接技术、**高密度组装技术**等。由于计算机最主要的硬件是集成电路，其集成度越来越高，功能越来越强，电路越来越复杂，所以集成电路的设计技术是计算机硬件设计的一个主要内容。多层印制电路板的设计和制造，特别是多层布线及连接各层的金属化孔的设计制造是计算机硬件的另一项重要技术。高密度组装应考虑元器件之间和线间的干扰以及通风散热等。

计算机硬件测试包括**元器件测试**、**印制板测试**、**电源测试**、打印机和磁盘驱动器等设备的测试、整机测试等。传统的检测工具，如电工仪表和示波器，仍在使用。专门的检测设备，如逻辑分析仪和监测器，已普遍被采用。有些计算机有自检电路，能自动发现一位或多位错误，甚至能自动校正错误。硬件故障可由软件检测，已开发出多种检测程序和诊断程序，可将故障定位在集成电路芯片上。

计算机硬件的可靠性包括**元器件可靠性**、**元器件老化**和**筛选**、**加固技术**、**防信息泄漏技术**、**电磁兼容性**等。为防止元器件的早期损坏对计算机可靠性产生影响，在组装前要对元器件进行老化和筛选。在对可靠性要求特别高的设备中，可使用容错技术和冗余技术。处于运动载体（车载、舰载、机载或星载）的计算机要有抗震措施。在高温、高湿等恶劣

条件下的计算机要有防护措施。为了保密，要防止计算机自身的信息泄漏，还要采取措施防止环境的电磁波干扰。

计算机维护可以保证计算机系统的正常运行。它对计算机系统的运行效率、数据的安全与完整、延长计算机设备的使用寿命都有重要的作用。

计算机机房设施是保证计算机在机房中能安全可靠地运行而配置的各种设施。特别是大型计算机要求有良好的机房环境，包括适宜的温度和湿度、洁净度、抗静电和电磁干扰、完善的通风系统、不间断电源和良好的接地系统等。

发展展望

计算机将在高性能、大容量、小型化、低功耗、智能化等方面发展。未来计算机的功能将会更强，处理速度更快，存储容量更大，性能价格比更高，安全性更好，人机界面更友善，用途更广泛。**计算机网络**与**多媒体技术**的发展，不断改变人们的工作方式，提高人们的生活质量。计算机与通信技术的紧密融合，互相渗透，正在加速人类社会信息化的进程。

计算机的主要硬件是集成电路。多年来，集成电路技术按照摩尔定律发展，即集成电路的集成度（芯片单位面积所能集成的晶体管的数量）每 18 个月翻一番，性能也相应增加。到 2011 年，集成电路芯片上 1 mm^2 面积已可集成几百万个门电路，一个处理芯片上可集成几十亿个晶体管，运算速度已超过万亿（10^{12}）次每秒，单个动态存储器芯片的容量已达到 4 Gb（相当于 4×10^9 位），超级计算机的运算速度已超过千万亿（10^{15}）次。未来的存储器容量将进一步发展到万亿（10^{12}）位，集成电路的工作频率将从 GHz 量级发展到 THz 量级，数据传输速率将从 Gb/s 量级发展到 Tb/s 量级，超级计算机的运算速度将达到百亿亿（10^{18}）次量级。一些专用芯片（如数字信号处理器、图形处理器、计算机网络中的路由交通处理器等）的功能将越来越强，处理速度越来越快。

集成度指数增长带来的直接效益是晶体管成本的指数下降，导致各类计算机和数码电子产品的性能价格比越来越高，也越来越容易被社会和公众广泛接受，进而带动信息产业和应用的持续发展。随着计算机产品的日益普及和整个社会对绿色环保和可持续发展的期望日益增长，出现了强调节能和环保的绿色计算思想，对计算机硬件的要求也从单纯注重性能价格比，变成同时重视性能功耗比（也称性能每瓦特）。

随着微电子技术的进步，除了硅半导体器件继续发展外，锗硅异质结器件、砷化镓器件等也将会得到发展。同时，考虑到半导体器件进一步微小化面临的物理局限和巨大的工艺开发难度，摩尔定律将难以延续，人们已经开始开发和探索包括光器件、超导器件、量子器件、分子生物器件、纳米器件等在内的新型器件。

在信息存储技术方面，半导体存储器、磁存储器和光存储器的容量、存取速度和传输速率可望大幅度提高，而体积、功耗和价格却大幅度下降。一些新的存储介质和存储技术，尤其是以相变存储器为代表的非易失性存储介质和相应的存取技术，也正在研究并逐步走向实用。随着大数据时代的到来，海量数据的安全性、共享性和可管理性显得尤为重要，网络存储系统应运而生，其发展速度异常迅速，并取代了大部分传统的存储设备。

计算机输入输出设备的发展首先由单一形式发展成多种形式的综合；其次，在某些应用中，由计算机的附属设备发展成主导设备；再次，智能化程度更高。输入输出设备的未来发展将更加符合人体特征。人机交互的输入输出技术和设备将更好地支持文字、图形、图像、声音等多媒体信息的识别和融合，还将支持用户对环境的感知能力，包括多种传感技术、上下文感知技术等。随着移动计算的发展，输入输出设备还可实现无线互连。为了实现可折叠的电子报纸，显示屏幕将是可折叠的，由普通电池供电。

（夏培肃　张修　唐志敏）

jisuanji yingjian kekaoxing

计算机硬件可靠性（computer hardware reliability）　在规定的条件下和规定的时间内，**计算机硬件**保持其规定功能的能力（或概率）。规定的条件通常包括环境条件、使用条件、维修条件和操作技术。规定的时间是对产品规定的观察时间，包括连续使用、间断使用、储存和一次使用的时间。规定的功能是指产品的各项技术性能指标和失效判据。能力是在规定条件下和规定时间内，完成规定功能的程度。

计算机硬件可靠性包括**元器件可靠性**、设备可靠性和系统可靠性。可靠性技术包括对可靠性的评价、预测、分配、提高等。对于元器件要运用失效物理学来分析元器件失效的过程，找出这些过程与应力、环境条件和时间等各种因素的相互关系，确定失效模式和失效机理，并提出可靠性保证措施。对于

设备和系统要运用概率论和**数理统计**的理论和方法来研究其故障时间的分布、分布类型和分布参数，分析和预测影响可靠性的因素，提出评价计算机硬件可靠性特征的指标，以及计算、试验和分配方法，并提出计算机系统在设计、制造、试验和使用各阶段的可靠性保证措施。

可靠性技术的发展大致分为4个阶段：①调查研究阶段（1950—1957年），收集分析元器件的现场数据，研究寿命试验方法，成立专门的可靠性组织。②统计试验阶段（1957—1962年），研制环境和可靠性试验设备，开展产品统计抽样寿命试验，制定可靠性标准和管理规范，建立可靠性数据收集和交换系统。③可靠性物理研究阶段（1962—1967年），分析元器件失效机理，研究高可靠性设计和工艺，探讨加速寿命试验的方法。④可靠性保证阶段（1967年至今），提高整机可靠性，强化可靠性保证、管理、认证制度，形成质量保证体系，发展可靠性试验技术和改进可靠性标准。

根据环境和使用条件的情况，可靠性可以分为：①固有可靠性　指系统在设计、制造时的内在可靠性，包括原材料质量、电路设计的技术水平、机械结构和制造工艺等因素的影响。②使用可靠性　指使用人员和维护人员对系统可靠性的影响，包括使用人员的操作技术水平、维护人员的技术水平及其他各种人为因素。③储存可靠性　指设备或系统处于存放状态时，性能随着保持时间的增加所呈现的各种下降规律。④环境适应性　指系统所处的环境条件对于系统可靠性的影响，包括温度、湿度、振动、冲击、运输、烟雾、霉菌、辐射等。

可靠性及其定量指标是反映产品质量的综合指标，是产品从出厂开始到工作寿命终止全过程的一种特性，因而具有综合性、时间性和统计性的特点。可靠性定量指标有以下几种。

（1）瞬时失效率$\lambda(t)$　产品在t时刻后的时间间隔$[t, t+\Delta t]$内失效数与在t时刻还在正常工作的产品数之比，称为失效率函数，简称失效率。计算公式为

$$\lambda(t) = \lim_{\substack{N\to\infty \\ \Delta t\to 0}} \frac{n(t+\Delta t) - n(t)}{[N - n(t)]\Delta t}$$

式中，N为产品总数，$n(t)$与$n(t+\Delta t)$分别为t时刻与$t+\Delta t$时刻的产品失效数，$N-n(t)$为t时刻还在正常工作的产品数。失效率常用的基本单位为非特（fit），1 fit $=10^{-9}\mathrm{h}^{-1}$。产品典型的失效率曲线呈浴盆状，故又称浴盆曲线。

（2）可靠度$R(t)$　产品在规定的条件和规定的时间内，完成规定功能的概率，其计算式为

$$R(t) = \exp\left[-\int_0^t \lambda(t)\mathrm{d}t\right]$$

（3）不可靠度$F(t)$　产品在规定的条件和规定的时间内，丧失功能的概率。也称产品的失效分布或寿命分布或累积失效率，其计算式为

$$F(t) = 1 - R(t)$$

（4）失效密度函数$f(t)$　产品在t时刻后的单位时间内的失效概率，其计算式为

$$f(t) = \mathrm{d}F(t)/\mathrm{d}t = \lambda(t)R(t)$$

（5）平均寿命$E(t)$　对于不可修复的产品，是指失效前的平均工作时间或储存时间，即平均无故障时间（MTTF）；对于可修复的产品，是指两次相邻失效（故障）间的平均工作时间，即**平均故障间隔时间**（MTBF）。平均寿命的计算公式为

$$E(t) = \int_0^\infty tf(t)\mathrm{d}t$$

计算机硬件的可靠性技术包括避错技术和容错技术。避错技术的目的是尽量减少硬件故障发生的概率，减小硬件的失效率。容错技术是利用额外的硬件和时间两种冗余方式掩盖故障的影响。计算机硬件从下述三个层次提高其可靠性。

（1）提高元器件的可靠性　指选用高可靠度、高集成度的器件，并对上机的元器件进行可靠性筛选。同时改善环境条件，包括温度、湿度、振动、冲击等物理条件和限额使用，降低元器件的各种应力负载，延长使用寿命。

（2）提高装置级（插件级或功能部件级）的可靠性　指印制板布线、印制板生产、电气装配等生产管理和质量管理。

（3）提高系统可靠性　包括冗余结构设计（并行、备用、故障检测、故障屏蔽、动态冗余等），缩短修理时间（自动故障诊断，提高可维性等），系统的自动恢复技术等。

产品的可靠性取决于**可靠性设计**、元器件优选和**元器件筛选**以及生产加工工艺三个环节，其中可靠性设计是最重要的。

参考文献

1. 傅佩琛，赵霖，张军英. 计算机系统硬件软件可靠性理论及其应用. 北京：国防工业出版社，1990

2. 猪濑·博. 计算机系统的高可靠性技术. 尤国峻，肖俊远，译. 北京：国防工业出版社，1985

（孟凤珍）

jisuanji youxi

计算机游戏(computer games) 在计算机上运行的一种具有娱乐功能的游戏软件。

计算机游戏的发展与**计算机图形学**、**计算机动画**、**人工智能**等密切相关。计算机游戏产业的市场惊人,可以与电影产业相媲美。计算机游戏的出现与20世纪60年代计算机进入美国大学校园有密切的联系。当时的环境培养出了一批编程高手。1962年大学生斯蒂夫·拉塞尔在美国DEC公司生产的PDP-1型计算机上编制的《宇宙战争》(Space War)是当时很有名的计算机游戏。一般认为,他是计算机游戏的发明人。1971年,被誉为“电子游戏之父”的诺兰·布什纳尔发明了第一台商业化电子游戏机。进入20世纪90年代,计算机软硬件技术的进步和因特网的广泛使用为计算机游戏的发展带来了强大的动力。计算机游戏按平台可以分为单机游戏和网络游戏。由于网络游戏中玩家的对手是人,而不是单机游戏中的计算机,因而更具有吸引力。进入21世纪,网络游戏成为计算机游戏的一个新的发展方向。

网络游戏的形式有很多种类,但都离不开战略游戏(realtime strategy game)、动作游戏(action game)和角色扮演游戏(role playing game)。在战略游戏(RTS)中,一切都是实时发生的,要求玩家具备较好的敏捷与宏观指挥能力。代表性的战略游戏有《红色警戒》(Red Alert)。动作游戏强调玩家的反应能力和手眼的配合。动作游戏的剧情一般比较简单,只要熟悉操作技巧就可以进行游戏。通常要求玩家所控制的主角(人或物)根据周围所遭遇的情况变化做出一定的动作,如移动、跳跃、攻击、躲避、防守等,来达到游戏所要求的目标。动作游戏讲究逼真的形体动作、火爆的打斗效果、良好的操作手感及复杂的攻击组合等。一般比较有刺激性,情节紧张,声光效果丰富,操作简单。动作游戏还可进一步分为射击游戏(shooting game)和格斗游戏(fighting game)。射击游戏(STG)代表作品有《彩虹六号》(Rainbow 6),格斗游戏(FTG)代表作品的有《热血街霸》(GetAmped)。

角色扮演游戏(RPG)是目前最受玩家欢迎的游戏类型之一,作为一种新兴的娱乐平台,其发展前景广阔。角色扮演游戏提供玩家一个可供冒险的世界或者一个反映真实的世界,这个世界包含各种角色、建筑、商店、迷宫及各种险峻的地形。玩家扮演虚拟世界中的一个或者几个特定角色在特定场景下进行游戏。玩家所扮演的角色在这世界中通过旅行、交谈、交易、打斗、成长、探险及解谜来揭开一系列的故事情节线索,最终走向胜利的彼岸。玩家依靠自身的胆识、智慧和机敏获得一次又一次的成功,使自己扮演的主角不断发展壮大,从而得到巨大的精神满足。代表性的作品有《传奇》、《A3》、《魔兽世界》等。

游戏的开发离不开游戏引擎。游戏引擎是用于控制游戏功能的主程序,如接受玩家控制信息的输入,选择合适的声音以合适的音量播放等。引擎相当于游戏的框架,框架搭好后,关卡设计师、建模师、动画师可往里填充内容。在3D游戏的开发过程中,引擎的制作往往会占用非常多的成本。游戏引擎提供多种功能,如场景的光照效果、角色的动作设计、绘制、人机交互、物理系统等。物理系统使游戏场景中物体的运动遵循物理规律,使得运动具有真实感。碰撞检测是物理系统的核心部分,它检测游戏中各物体的物理边缘,以防止两个三维物体撞在一起时相互穿过。当玩家撞在墙上的时候,碰撞检测程序会根据玩家和墙之间的相对位置确定两者的相互作用。

在游戏机产业中,主要产品有微软的Xbox、Sony公司的PlayStation2和任天堂的GameCube。

参考文献

1. 耿卫东,陈为. 计算机游戏程序设计. 2版. 北京:电子工业出版社,2009

2. 彭群生,金小刚,万华根,等. 计算机图形学应用基础. 北京:科学出版社,2009

(耿卫东)

jisuanji zhengji jiance

计算机整机检测(computer hardware system testing) 对运行在设计技术指标所规定的条件下的计算机整机进行的综合性能指标的检查和测试。检测的主要内容包括:整机图纸、文档审查以及系统速度、系统正确性、系统响应能力、系统可靠性、系统可用性、系统可维性、系统稳定性及系统友善性等。进行整机检测的主要目的是对被检测的计算机作出较全面的鉴定和评价,帮助计算机研制、生产厂家改进其产品的性能和质量,帮助计算机用户选购适用的计算机或改进现有计算机的性能。

进行计算机整机检测时,首先,应使被测整机运行在设计技术指标所规定的温度、湿度、大气压强、供电电源、工作频率等环境条件之中。其次,必须选择经过权威计量管理部门鉴定和确认并适用于被测

计算机的仪器设备。再次，应依据被测机的特点和用户需求选取或编写并经验证的基准测试程序集和检测用例。最后，要制订详细的检测计划。对在检测过程中可能出现的情况，如因偶然事件（停电、更换操作员等）中断测试的处理、故障类型确认、故障时间判定、故障记录及故障排除等，作出明确的规定。总之要尽可能地保证检测的顺利进行和获得高可信度的检测结果。

整机图纸、文档审查　整机图纸、文档审查是依据被测计算机应当遵从的标准（国标、军标、部标等）检查其图纸、文档是否齐全，是否符合标准，还要通过实际考查被检测计算机的主要技术指标，如工作频率、基本字长、处理器类型、存储器容量、系统结构、实际配置、扩展能力、工作环境条件等，验证是否符合设计要求，从而使检测人员了解被测机器的技术水平，为其在随后的测试中如何掌握标准提供技术依据。

系统速度测试　包括系统峰值速度、系统持续速度和系统估算速度等测试方法。它们各自从不同的角度反映整机系统的速度指标。

（1）系统峰值速度　系统峰值速度 v 是被测系统使用的处理器芯片的标称速度 v_e 同该系统所含执行运算的这种处理器的数量 n 的乘积，亦即

$$v = v_e n \tag{1}$$

（2）系统持续速度　用适合被测系统特点的基准测试程序集，分别在已知标称持续速度的参考机和被测机上运行，测得各个基准测试程序在这两种机器上运行完成所需的时间，而后算出被测机器的系统持续速度，其计算公式为

$$v_a = v_b \sqrt{\prod_{i=1}^{n} \frac{T_{bi}}{T_{ai}}} \tag{2}$$

式中，v_a——被测机系统持续速度；

v_b——参考机标称系统持续速度；

n——基准测试程序集中所含的基准测试程序数；

T_{bi}——在参考机上运行完成第 i 个基准测试程序所需的时间；

T_{ai}——在被测机上运行完成第 i 个基准测试程序所需的时间。

（3）系统估算速度　选择被测计算机主要应用领域有代表性的课题集，令其在被测机上运行，记录各个课题运行完成所需的时间，再根据各个课题的程序执行量，算出平均执行速度，即为系统估算速度，其计算公式为

$$\bar{v} = \frac{n}{\sum_{i=1}^{n} T_i / q_i} \tag{3}$$

式中，$\bar{v}$——系统估算速度；

n——选用课题集包含的课题数；

T_i——运行完成第 i 个课题所需的时间；

q_i——第 i 个课题的程序执行量。

上述三种系统速度测试方法，以系统持续速度测试法最通用，所测得的结果也有较高的可信度和较强的说服力。

整机正确性检测　检测计算机解题的正确性。进行这种检测，首先要依据下述原则选择测试课题：①主要应用领域有代表性的各类课题；②典型算法和常用算法的各类课题；③高效、中效、低效的各类课题。然后，令各个课题在被测机上运行三次，每次运行之间有一定的时间间隔。记录每次运行的结果，进行比较，若三次运行结果一致并达到被测机设计精度或同已知的正确结果一致，则认为被测机通过了正确性检测。

系统响应能力检测　系统响应能力检测包括系统响应时间 t 和系统周转时间 T 检测。

（1）系统响应时间　系统在交互方式和其设计允许的负荷条件下，自用户终端键入命令的时刻到系统在此终端上输出响应该命令的第一个字符的时间。然而，由于计算机有多种命令，其功能各异，各自的系统响应时间相差较大，所以常用各条命令的系统响应时间的平均值表示。常用的检测方法是在终端上键入 n 条命令，记录每条命令的系统响应时间 $t_i(i=1,2,\cdots,n)$，然后用式

$$t = \frac{\sum_{i=1}^{n} t_i}{n} \tag{4}$$

算出系统响应时间。

（2）系统周转时间　系统在批处理方式和其设计允许的负荷条件下，自输入设备将各类作业提交给系统到其各类作业输出结果这段时间的平均值。检测方法是自输入设备将 m 个作业提交给系统，分别记录各个作业的系统周转时间 $t_i(i=1,2,\cdots,m)$，而后按式

$$T = \frac{\sum_{i=1}^{m} t_i}{m} \tag{5}$$

算出系统周转时间。

系统可靠性检测　系统可靠性（参见**计算机系统**

可靠性)常用系统的**平均故障间隔时间**(MTBF)表征。目前常用的检测方法是:根据被测计算机的设计所允许的负荷能力,选用适当数量的能够反映被测机特点及性能的正确性测试课题,使其同该机的综合诊断程序一起按某种方式在被测机上联合运行。运行时间至少应为被测机的 MTBF 设计指标的 3~5 倍的时间。其运行方式通常是正确性课题运行时间占整个运行时间的 1/2 以上,而综合诊断程序运行时间占 1/2 以下。在运行过程中详细记录系统故障、故障次数及修复时间等,运行达到确定的时间后,作出正确性报告和故障报告。在运算正确的条件下,算出被测整机的 MTBF,其计算公式为

$$\text{MTBF} = \frac{T_R}{K_S + 1} \tag{6}$$

式中,T_R——被测整机正确运行的时间总和;

K_S——被测整机在被测期间的系统故障次数。

系统可用性检测 系统可用性(参见**计算机可用性**)常用系统可用度 A 表示。系统可用度的测试方法同系统可靠性测试方法一样,可以使用在系统可靠性检测中获得的记录,算出系统的可用度,其计算公式为

$$A = \frac{T_R}{T_R + T_S} \tag{7}$$

式中,T_S——系统故障修复时间总和。

对于多机或多处理器或有多重部件的计算机系统,还可进行降级使用能力检测。它包括:①在被测系统停机时,对其可能降级使用的多种环境分别进行分割组合,然后令其运行测试课题,检验运行结果的正确性;②在系统运行状态下,通过人机通信切除或连接系统的某个部件,观察系统运行的变化和检验运行结果的正确性;③在被测系统运行状态下,人为地设置故障,由系统自行检出并切除故障部件,继续运行,完成后,检查运行结果的正确性。总之,必须在运行正确的前提下确认被测系统的降级使用能力。

系统可维性检测 系统可维性(参见**计算机可维护性**)常用**平均恢复时间**(MTTR)表示。故障恢复时间取决于维修人员技术熟练程度、维修的方便性、同类部件的互换性程度以及用系统的综合诊断程序进行故障检出和故障定位的速度及精度水平。这些影响系统可维性的因素最终都体现在平均恢复时间中,因而可以使用在系统可靠性检测中获得的数据算出被测整机的平均恢复时间,其计算公式为

$$\text{MTTR} = \frac{\sum_{i=1}^{K_S} t_i}{K_S} \tag{8}$$

式中,t_i——第 i 次故障的修复时间;

系统稳定性检测 计算机整机稳定性检测是计算机在其设计的环境指标的极限条件下能够正常运行程度的检测。常用的检测方法是对被测整机按某种标准施以各种环境极限条件,如振动、温度、湿度、电源拉偏、工作频率拉偏或电磁干扰等,令被测机在一段时间内运行基准测试程序或综合诊断程序。检验运行结果是否正确,并由此判断系统稳定性是否达到设计目标。

对微型和中型计算机的稳定性检测可以按照 GB 9813—1988 和 GB/T 137323—9 中的规定及其所引用的有关国家标准设置环境极限条件。对大型或巨型计算机的稳定性检测,目前尚无标准可循,常用的耐震能力测试是将整机拆开装车,在三级公路上运输 250 km 作为施震条件。对电源、工作频率常提供 ±5% 的拉偏范围。对抗电磁干扰能力的测试,可按国家标准 GB 9254,GB 6833 的有关规定提供环境极限条件进行。

系统友善性检测 系统友善性是计算机产品获得用户的重要条件之一。主要内容有系统使用方便程度、对误操作的系统防护和报警能力、求助系统的良好程度及可视性程度等。常用的检测方法是:先审查被测机有关系统友善设计的指标,同当时公认的友善性良好的机型相比,判别该机友善性设计完善程度。而后进行现场检测,通过实际编程、调试检验使用的方便程度;通过人为操作错误检验系统自身防护能力和报警能力;通过逐项求助验证求助系统的完善程度。然后通过荧光屏检验用户能直接看到系统的什么状态,如资源忙碌程度、作业调度情况、课题运行过程等。最后作出被测机的友善性评价。

以上检测内容使用了简易的采样检测方法和在计算机研制、调试、维修中常用的仪器作为检测工具。在大部分项目检测中,采用当前使用最多的综合基准测试程序集作为被测机的工作负荷。如此检测得到的结果能够对被测整机作出较全面的和较准确的评价。多年来人们为研究出具有普遍性和代表性的用于计算机检测工作负荷、事件驱动的软件、硬件监测器等检测工具作了大量的工作,取得了显著的进展。然而,研究出一种广泛适用的计算机检测标准规范,还需计算机设计、制造、应用等领域的人们共同努力。

参考文献

1. Domenico F, et al. 计算机系统的测量和优化. 郑衍衡,王春元,等译. 北京: 水利电力出版社,1988

2. 吴立德,吴霭成. 计算机系统性能评价. 上海: 上海科学技术出版社,1986

3. 傅佩琛,赵霖,张军英. 计算机系统硬件和软件可靠性理论及其应用. 北京: 国防工业出版社,1990

(程乐祥)

jisuanji zhichi de xietong gongzuo

计算机支持的协同工作(computer supported cooperative work,CSCW) 地域分散的群体借助计算机及其网络技术的支持相互协调与协作来完成一项任务的技术领域。它包括协同工作体系结构、群体协作方式、模型和支持群体工作的相关技术、应用系统的开发等部分。通过建立协同工作的环境,改善人们进行信息交流的方式,消除或减少人们在时间和空间上相互分隔的障碍,节省工作人员的时间和精力,提高群体工作质量和效率,从而提高企业、机关、团体,乃至整个社会的整体效益和人类的生活质量。

计算机支持的协同工作(CSCW)的概念是1984年被正式提出来的。为了使群体的成员能协调地配合工作,参加协同的人员需要关于其他协作者必要的共享信息;能把任务在参与任务的成员之间分解成单元,完成以后能重新组合成一个整体;任务参与者和整个群体都还需要使技术能适应各自的实际情况。这些需求只有在计算机和网络技术的支持下才有可能实现,因此它是一个多学科交叉的研究领域。在实际应用中,除了需要计算机网络、通信技术、多媒体技术、分布式计算机系统等计算机技术的支持以外还需要社会学、心理学、管理科学等多个领域学者共同协作,从而向人们提供了一种全新的工作环境和交流方式。

计算机支持的协同工作的分类

计算机支持的协同工作(CSCW)的分类群体协作方式的多样性,为CSCW研究提供了丰富的内容。在CSCW系统中,人们围绕着共同的任务需要进行交互通信、协调、协作和协同等基本活动,可以根据CSCW系统中的基本活动方式、群体成员地理分布位置、使用的基本工具和工作环境、应用等对CSCW系统进行分类:

按群体成员之间的交互协作方式可分为同步方式和异步方式两种。

按群体成员的地理分布可分为同地协作和异地或远程协作。

按群体规模可分为两人协同系统和多人协同系统。

按使用的基本工具和工作环境可分为信报系统,即**电子邮件**系统,公告板系统、会议系统、协同写作和讨论(编著)系统、工作流系统和群件等。

按CSCW应用系统可分为协同科研系统、协同设计系统、远程医疗系统(参见**医疗信息系统**)、远程教育系统(参见**计算机辅助教学系统**)、协同决策系统、军事协同(参谋会议)系统和协同办公系统等。

按照上述各种分类的观点,可以把各种CSCW系统构成如图1所示的一种立体模型。

计算机支持的协同工作系统体系结构

计算机协同工作(CSCW)系统典型的体系结构可以表示为如图2所示的一个4层结构的模型。第1层为“开放系统互连环境”,提供开放的通信网络支持环境,保证协同工作过程中有效的信息交流。第2层为“协同工作支撑平台”,解决协同工作所需的主要机制和工具。主要机制如信息共享、信息安全控制、群体成员管理,基本工具包括电子邮件系统、会议系统、协同写作和讨论系统、工作流系统等。第3层为“协同工作应用接口”,在这一层中需要提供协同应用的编程接口(API)、人机交互(HCI)和智能外围接口(IPI)。通过标准化的服务接口向应用系统提供第3层的功能,使上层的应用系统与下层的支撑平台具有相对的独立性;提供有效、灵活、方便的人机交互接口以及在协同工作环境下协作各方交互关系、规则和策略等。第4层为“各种CSCW应用系统”,针对各种协同工作应用领域,提供所需的协作支持工具的剪裁和集成以及协同应用系统的开发。

计算机支持的协同工作的关键技术

CSCW关键技术基础是计算机及计算机网络技术,主要的发展动力来源于广泛的应用需求。关键技术的深入研究是CSCW应用系统飞跃发展的基础。它们包括: CSCW系统模型和体系结构、群体协作模式、协作控制机制、CSCW系统中的群组通信支持、多媒体技术、多重交互接口(人际交互、人机接口、应用编程接口)技术、应用共享技术、CSCW系统的安全控制、CSCW应用系统开发环境和应用系统集成技术等。

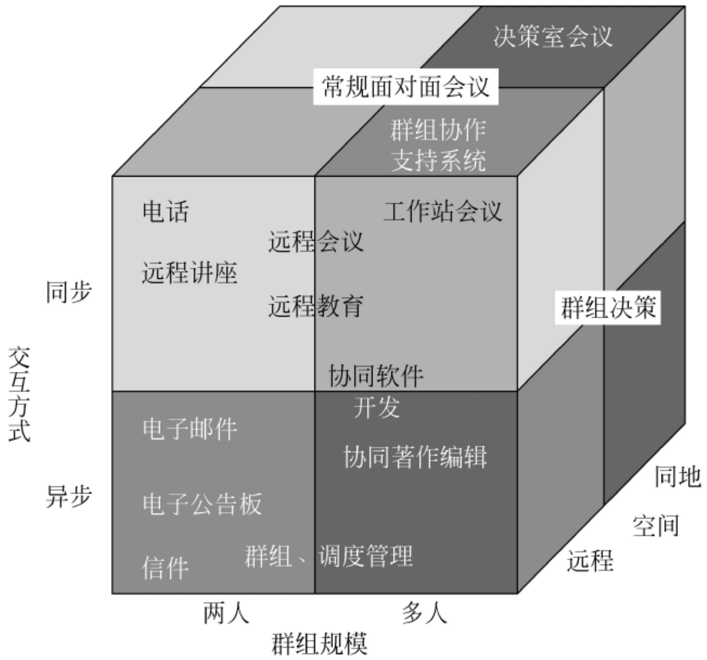


图 1　计算机支持的协同工作(CSCW)基本系统的分类



图 2　计算机支持的协同工作(CSCW)
系统体系结构

计算机协同工作技术的应用领域

CSCW 具有十分广泛的应用领域,例如,军事应用、工业应用、协同计算机辅助设计(CO-CAD)、办公自动化(OA)和管理信息系统(MIS)、远程医疗、远程教育、合作科学研究、电子商务和商业、贸易、金融领域中的应用、电子政务和各级政府部门的协调和决策支持等,几乎包括了人类社会生活、生产的所有领域。

参考文献

史美林. 计算机支持的协同工作:理论和应用. 北京:电子工业出版社,2000

(史美林　徐光佑)

jisuanji zucheng

计算机组成(computer organization)　计算机及其各个主要功能部件(**中央处理器**、存储器、输入输出设备以及各部件之间的互连)的组成、功能及其实现。计算机通过执行程序可以完成计算、模拟、控制、辅助设计和事务处理等工作。**程序**是由指令组成的,因此计算机执行程序的过程实际上就是按照一定的次序执行一系列指令的过程。计算机从**输入设备**接收程序和数据,存放在存储器中,在中央处理器的控制和参与下处理数据,完成每条指令的操作,最后将结果数据通过**输出设备**输出。

计算机的基本组成

1945 年冯·诺依曼等人提出存储程序的计算机方案,其后各国研制的计算机基本上采用这种方案,称为冯·诺依曼计算机或存储程序计算机。直到今天,虽然计算机体系结构已有很大的发展,但仍可认为大部分计算机是按照冯·诺依曼计算机的基本原理组成的。早期的冯·诺依曼计算机的主要特点为:①采用存储程序方式,程序和数据存在同一存储器中。②存储器结构是按地址访问的线性编址的一维结构,它的字长是固定的。③数据以二进制表示。④机器以运算器为中心,输入输出设备和存储器之间的数据传送都经过运算器。⑤指令由操作码和地址码组成,一般按它在存储器中的存放顺序执行,程序的分支由转移指令实现。图 1 是以运算器为中心的计算机组成。这种由运算器、控制器包揽一切指令操作和输入输出操作的组成,使计算机必须等待正在执行的输入输出操作完成后才能进行

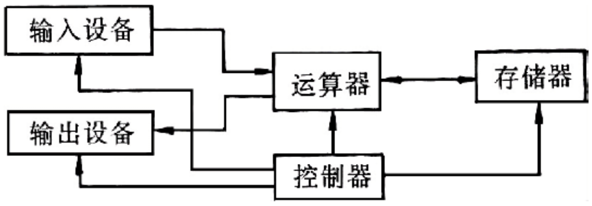


图 1　以运算器为中心的计算机组成

下一操作,影响了计算机的运行效率,代之而起的是以存储器为中心的计算机组成,如图 2 所示。批量的输入输出数据可以直接在输入输出设备和存储器之间进行传送。只是在必要时才用中断处理方式请求运算器和控制器进行干预。后来将运算器和控制器统一称为中央处理器。随着元器件的改进,尤其是集成电路的集成度从小规模逐步发展到超大规模,使中央处理器的体积逐步从几个机柜缩小到 1 个插件板,最后集成在 1 个芯片内。现在甚至可将浮点处理器、高速缓存和存储控制部件集成在中央处理器芯片内。

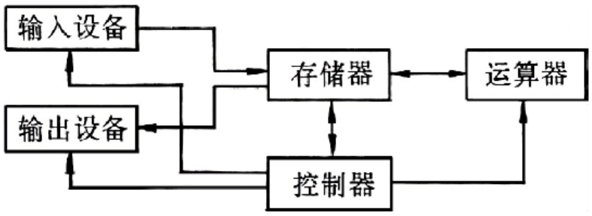


图 2　以存储器为中心的计算机组成

下面分别介绍计算机的基本组成部件。

中央处理器(CPU)　中央处理器包括运算器和**控制器**。

1. 运算器

运算器至少包括:①完成定点数算术运算和逻辑运算的算术逻辑运算单元 ALU(参见**算术逻辑部件**);②完成移位功能的移位器(也可以在 ALU 中完成移位操作);③提供操作数或保存中间运算结果的**通用寄存器**或专用寄存器。

算术逻辑部件一般不包括独立的乘法器和除法器,乘法和除法运算可通过多次交替执行加法(或减法)运算和移位操作来实现,早期的小型计算机和微型计算机甚至连这种功能也没有,而由子程序来完成乘法和除法运算。功能较强的中央处理器内可设置专门的高速乘法器和除法器,还允许它与 ALU 并行工作,以提高计算机的运算速度。浮点运算可根据计算机应用的要求,用程序实现或由浮点运算部件实现。过去浮点部件不包括在 CPU 芯片内,而有专门的浮点运算芯片,但随着器件集成度的提高,到 20 世纪 80 年代末和 90 年代初,在 CPU 芯片内可包括浮点运算部件。为了提高运算速度,在中央处理器内除了设置多个能同时并行工作的功能部件以外,每个功能部件还可以采取流水线方式工作。

早期的计算机执行算术逻辑运算指令时,其中一个操作数往往由运算器中的累加器提供,另一操作数在存储器中,运算结果也放在累加器中。后来的计算机一般设置多个通用寄存器,它们既可提供操作数和保存操作结果,也可提供各种**寻址方式**所需的基址、变址、主存地址或保存调用程序的参量和堆栈指针等。除此之外,还可设置诸如暂存寄存器、条件码寄存器等专用寄存器。精简指令集计算机(RISC)内设置有数量相当多的通用寄存器,这种计算机只有存数取数指令才访问存储器,其他操作都在寄存器之间进行,因此在采用流水线技术后,大多数指令能在一个机器周期内完成。

堆栈计算机用存储器中的堆栈来替代通用寄存器,但其栈顶和靠近栈顶的一些单元内容可能用寄存器存放,对于一般的双操作数算术逻辑运算,采用零地址指令对栈顶的两个操作数进行运算,结果送回堆栈。中央处理器内还设置一些寄存器用来保存主存中程序区、数据区和堆栈的各种指针。

2. 控制器

中央处理器中的控制器主要是指控制指令流和控制每条指令执行的指令控制器,由它产生执行各类指令操作所需的时序控制信号。这些控制信号可以用硬连线逻辑(又叫组合逻辑)(参见**硬连线控制器**)产生,也可以用微程序控制方式(参见**微程序控制器**)产生。两者相比,前者能获得更快的指令执行速度,后者更适宜于复杂指令控制信号的设计。有些计算机控制器是由这两种方式结合构成的。为了获得尽可能高的指令吞吐率,可以采用指令重叠执行的流水线技术;或采用先行控制的方法提前从存储器取出指令进行预处理,当遇到执行时间较长的指令时,尽可能把条件具备的后续指令提前执行;也可以设立指令缓冲站,以减少从主存储器重复读取同一指令的时间。

中央处理器除完成运算控制和指令控制功能外,还可通过**中断**系统调用**操作系统**,对存储器系统和输入输出设备进行统一管理;也可通过中断系统实现实时处理。

存储器　一般指的是**主存储器**。中央处理器可

直接访问它，外围设备也频繁地和它交换数据。存储器的存取速度往往满足不了中央处理器的快速要求，它的容量也满足不了用户解题的需要，为此提出了多层次存储系统和**虚拟存储器**的概念。存储系统可分为3个层次，它们是：高速缓冲存储器cache(简称高速缓存)(参见**高速缓冲存储器**)、**主存储器**和**辅助存储器**。为了提高存储器的存取速度，在主存储器与中央处理器之间增加了高速缓存；为了扩大存储器容量，使用磁盘存储器等作为辅助存储器(主存储器的后援)。用户使用时，可以面向辅助存储器的逻辑地址空间，从而使存储器达到高速(接近于高速缓存的速度)、大容量(接近于辅助存储器的容量)的要求。数据和指令在高速缓存与主存储器之间调动是由硬件自动完成的；在主存储器与辅助存储器之间调动是在硬件和操作系统控制下自动完成的。

在某些计算机的存储器中还包含少量只读存储器，用来存放一些操作所必需的基本程序，例如引导程序和基本输入输出系统BIOS等。

存储控制器实现对存储器的读写控制，它接收来自中央处理器或输入输出设备、辅助存储器的读写请求，向存储器发读写命令。并对同时来的读写请求，根据优先级进行仲裁。另外，还要控制动态存储器进行“刷新”操作。

外围设备及其控制 辅助存储器和输入输出设备统称为**外围设备**。低速的外围设备通过中断系统与**主机**(中央处理器和主存储器)交换数据。高速外围设备直接与存储器交换成批数据，称为直接存储器存取DMA(参见**直接存储器存取**)，但在一批数据传送前或传送结束后或传送过程中产生故障时，仍然需要通过中断程序予以处理。

在外围设备数量较多的情况下，全部设备都与存储器分别独立相连是困难的，而且也不能适应进一步扩充数量和更换品种的需要。为了适应计算机系统灵活多变的要求，现代计算机普遍采用总线互连方式，将所有的外围设备经过统一的总线(称为**系统总线**或输入输出总线)和中央处理器、存储器相连接，如图3所示。所有的外围设备都经过各自的控制器(称为接口)连接到总线上。为了增强通用性，总线的标准化工作很重要，例如ISA，EISA，Multibus，VME，PCI，NuBus和Future Bus等均为标准总线。由于各部件间传送信息都是通过总线进行的，而在同一时间内，一组总线只能为一对功能部件服务，因此，当同时存在两个或两个以上的传送要求

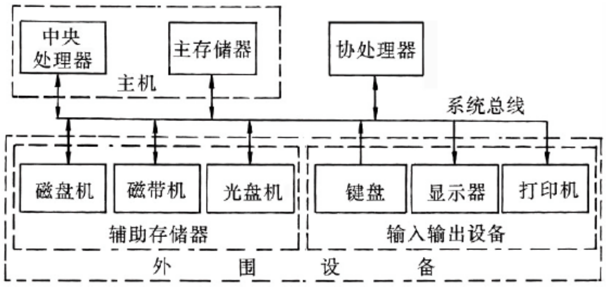


图3 以总线互连的计算机组成

图中的协处理器可以是浮点部件、数组处理机或图形处理加速器等。

时，需要根据优先级进行排队，因而总线有可能成为系统的瓶颈。在设备数量多和速度要求高的情况下还有其他连接方式，例如在主机内增设高速总线，使中央处理器与主存储器之间的传送通过高速总线进行；或者设置通道处理机(或输入输出处理机)将所有的外围设备管起来，分担主机的部分工作。

计算机中常用的逻辑电路

中央处理器、存储控制器和输入输出接口部件基本上是由各种逻辑电路构成的。在计算机中常用的逻辑电路可分为组合逻辑电路和时序逻辑电路两大类。

1. 组合逻辑电路

这种电路和输出状态仅和当时的输入状态有关，而与过去的输入状态无关。常用的有加法器、译码器和数据选择器等。构成这些电路的基本单元是“与”门、“或”门和反相器，通常可用逻辑表达式(或用真值表)来表示其功能。

(1) 加法器 一位加法器可对两个相加数中的某一位以及从低位来的进位数相加，产生本位运算结果以及向高位的进位数。32个加法器可实现两个32位数的加法运算，但进位数延时较长，因此通常附加有快速进位电路。加法器是构成运算器的主要单元。

(2) 译码器 译码器有 n 个输入变量，$2n$ 个输出信号，当输入为某一组值时，仅有对应的一个输出为“1”(或“0”)，其余均为“0”(或“1”)信号。译码器在计算机中得到广泛的应用，如指令译码器、存储器中的地址译码器等。

(3) 数据选择器 从多个输入通道中选择一个通道的输入数据作为输出数据。

2. 时序逻辑电路

这种电路的输出状态不但和当时的输入状态有关，而且还与在此之前的输入状态有关。时序逻辑

电路内必须要有能存储信息的元件(称为触发器)。触发器是构成时序电路的基本单元,用触发器和控制门可组成寄存器和计数器等,在计算机中应用的有指令寄存器、数据寄存器、缓冲寄存器、移位寄存器和程序计数器等。

另外,由基本门电路组合而成的可编程逻辑器件(PLD)在计算机中经常被采用,PLD 由两级门阵列电路构成,第一级是“与”阵列,第二级是“或”阵列,根据“与”阵列和“或”阵列是否可由使用者编程又有可编程只读存储器(PROM)、可编程逻辑阵列(PLA)、可编程阵列逻辑(PAL)、通用阵列逻辑(GAL)和**现场可编程门阵列**(FPGA)等。

现代计算机组成特点

单处理器向提高主频和增强功能方向发展 由于芯片集成度和速度的提高,现代微型计算机几乎已具有过去大型计算机的所有特点:

(1) 运算部件的多功能化　微处理器芯片内可含有多个 ALU 部件、乘法器、除法器、地址部件(形成访存地址)和浮点部件等。

(2) 流水线　包括多条指令重叠工作的指令流水线和多个数据重叠处理的运算流水线。指令相关、数据相关、转移指令和中断处理会影响流水线效率,为此可采用编译优化、数据旁路、转移预测等方法进行改进。

(3) 高速缓存　微处理器芯片内含有第一级指令高速缓存和数据高速缓存(共用或分开),片外可根据需要连接容量更大的第二级高速缓存。某些微处理器甚至考虑了三级高速缓存方案。某些辅助存储器(例如磁盘存储器)也具有高速缓存。

(4) **虚拟存储器**　设置存储管理部件(MMU),将逻辑地址转换成物理地址,并与操作系统配合,自动完成主存储器与辅助存储器之间的数据传送。有的微处理器已将原来在芯片外的存储管理部件移到片内。

(5) 多处理器　微处理器芯片内可含有适应多个处理器一起工作的指令和相应逻辑,甚至实现了在单个芯片内集成多个处理器核的产品。

(6) 接口标准化　使系统便于升级。

单机向多机并行处理系统发展 现代微处理器芯片的主频将很快接近其极限,因此依靠提高主频来提高机器性能的潜力已不很大,更有效地提高性能的途径是改进系统结构,提高并行度。

在科学研究和工程设计的某些应用领域,需要对巨大的数组进行计算,一般的通用大型机不适合于这种大量的数值计算,为此,发展了向量流水线计算机(称为向量处理机)。

阵列处理机是一种并行处理系统,它有 1 个指令部件和多个数据处理单元。在指令部件控制下,所有的数据处理单元对分配来的数据并行地完成 1 条指令所规定的操作。

多处理机系统(参见**并行处理系统**)将很多个处理器组合在一起构成一个计算机系统,速度可以很高,价格相对来说比较便宜。

集中式处理系统向分布式处理系统发展 将地点分散的、具有独立工作能力的多台计算机通过通信网络相互连接起来,在系统软件控制下,实现资源共享并共同完成一个任务的系统,称为分布式计算机系统。连接计算机的通信网络可以是**局域网**、**城域网**、广域网或**互联网**。所连接的计算机可以是同种机型的或异种机型的(参见**分布式异构型计算机系统**)。客户-服务器系统(参见**客户-服务器计算**)、计算机簇(参见**集群计算**)等都是分布式计算机系统。

开放系统(参见**开放系统**)　开放系统指计算机体系结构、系统总线、操作系统、窗口系统、数据库、图形用户接口、计算机网络和通信服务等都是开放的,符合国际标准或工业标准。这样,硬件和软件厂商就很容易进行分工,他们的产品也能很方便地集成在一起工作;用户可以选用市场上最适合于他们的软件和硬件组成计算机系统。因此,具有开放系统特点的计算机系统之间有着良好的可移植性和互操作性。可移植性是指在一个计算机系统上运行的操作系统(或应用软件)可以很容易移植到另一计算机系统上;互操作性是指不同厂家的各类计算机可以相互交换信息,为达到互操作性要求,各厂家的计算机应符合统一的通信协议、数据格式等,而且具有良好的互连性。

参考文献

1. 王爱英. 计算机组成与结构. 4 版. 北京:清华大学出版社,2007

2. 孙强南,孙昱东. 计算机系统结构. 北京:科学出版社,1992

(王爱英)

jisuan jihe

计算几何(computational geometry)　理论计算机科学的一个分支,主要研究解决几何问题的算法。人们于 20 世纪 70 年代初已尝试用计算机计算平面点集的凸壳(或称凸包)。而到 70 年代中期,

沙莫斯(Shamos)和霍伊(Hoey)利用计算机成功地计算平面点集的 Voronoi 图,并发表了一篇著名论文,该论文命名了一门新的学科——计算几何。

计算几何的研究成果应用于计算机图形学、地理信息系统、路径规划、模式识别、模具加工、分子建模、一维和二维下料、CAD/CAM 等诸多领域。

凸壳、Voronoi 图、点集的三角剖分、多边形的中轴、几何体的划分与等分、交与并、点的定位,区域查找等诸多问题都是计算几何研究的问题,并且随着时间的推移,研究的范围将越来越广泛。总之,计算几何是研究求解几何问题(包括可转化为几何问题的问题)的算法、算法复杂性、问题复杂性的一门学科。这里,仅介绍凸壳、Voronoi 图、点集的三角剖分、多边形的中轴等基本内容。感兴趣的读者请见参考文献。

凸壳

平面点集 S 的凸壳边界定义为一凸多边形,该凸多边形的顶点均为 S 中点,并且 S 中没有一个点位于该凸多边形的外部。凸壳由凸壳边界及其内部区域构成。点集 S 的凸壳记为 CH(S)。此外,利用凸集的概念亦可以定义凸壳。利用凸壳的概念可以简化许多问题的求解。例如,点 P 到点集 S 的距离($\min_{P_i \in S} d(P,P_i)$)可以用点 P 到 CH(S)的距离来代替。

计算平面点集 S 的 CH(S)就是从 S 中寻找 CH(S)边界的顶点并按序将其连接。由于计算 CH(S)问题与排序问题关系密切,故可以由排序问题的下界推得计算 CH(S)问题的下界,它们均具有相同的下界 $O(n\log n)$,其中 n 为 S 中点的数目。计算 CH(S)的常用算法有卷包裹法、格雷厄姆(Graham)方法、分治法、$Z_{3\text{-}1}$ 算法、$Z_{3\text{-}2}$ 算法、实时凸壳算法、增量算法、近似凸壳算法等。另外,计算三维空间中的 CH(S)亦有一些算法。除串行凸壳算法外还有并行凸壳算法以及时间复杂性低于 $O(n\log n)$ 的凸壳算法。

计算点集 S 的 CH(S)往往是求解某些问题的预处理工作。例如,寻求点集的直径,货郎担问题的求解等。如果事先计算点集的凸壳,那么将为后继计算带来便利并可降低计算的时间复杂性。

Voronoi 图

Voronoi 图与凸壳同为计算几何中的重要几何结构。利用 Voronoi 图可以求解几何对象与距离有关的问题。例如,寻求平面点集中最近点对、最大空圆、最小生成树等。

给定平面点集 $S=\{P_1,P_2,\cdots,P_n\}$,关联于点 P_i 的 Voronoi 多边形(或 Voronoi 域)定义为

$$V(P_i) = \bigcap_{i \neq j} H(P_i,P_j) \qquad (1)$$

其中 $H(P_i,P_j)$ 表示,线段 $\overline{P_iP_j}$ 的垂直平分线 L,L 划分平面为两个半平面,P_i 所在的半平面即为 $H(P_i,P_j)$。利用式(1)可以计算 $V(P_i)$,然后计算 S 中每个点的 Voronoi 多边形,便得到点集 S 的 Voronoi 图,记为 Vor(S)。图 1 中实线为 Vor(S)。该图中的顶点和边分别称为 Voronoi 顶点和 Voronoi 边。Vor(S)中,有些 Voronoi 域是无界的。产生 Voronoi 边的线段构成 Vor(S)的对偶图,如图 1 中虚线所示。

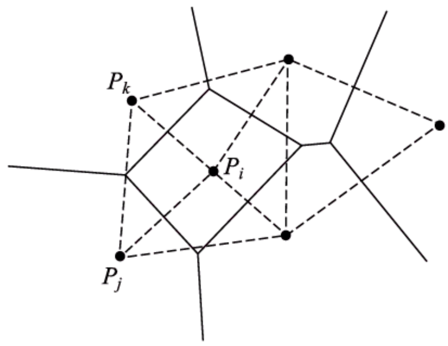


图 1 Voronoi 图及其对偶图

n 次利用式(1),可以得到 Vor(S),其时间复杂性高于 $O(n^2)$。此外,构造 Vor(S)的算法还有增量算法、分治算法、平面扫描算法、$Z_{4\text{-}1}$ 算法等。前者时间复杂性为 $O(n^2)$,其他是 $O(n\log n)$。

高阶 Vor(S)以及 Vor(S)与 CH(S)的关系已有较好的算法及结果。除点集 S 的 Vor(S)外,S 还可以是线段集、点线集、多边形集等。

三角剖分

给定平面点集 $S=\{P_1,P_2,\cdots,P_n\}$,所谓 S 的三角剖分是指用互不相交的直线段连接 P_i 与 P_j,$1 \leq i,j \leq n, i \neq j$,并使 CH($S$)内的每个区域是一个三角形。三角剖分是一个平面图,它有 n 个顶点,$3n-3-k$ 条边及 $2n-2-k$ 个三角形,其中 k 为 CH(S)顶点数(点处一般位置)。对三角剖分问题可以提出不同的约束条件,从而产生诸多的三角剖分问题。例如,最小权三角剖分(即边的总长度最小化)、最小角最大化等。

点集 S 的最近点意义下 Vor(S)的对偶图是点集 S 的一种三角剖分,即 Delaunay 三角剖分,记为

DT(S)。最远点意义下 Vor(S)的对偶图是 CH(S)内区域的一种三角剖分。

计算点集 S 的三角剖分有一些算法:贪心算法、$Z_{4\text{-}3}$算法等。由于点集的最小权三角剖分问题是 NP 难的,故不存在计算点集最小权三角剖分的多项式时间算法。

除点集的三角剖分问题之外,周培德提出了线段集、点线集三角剖分问题并进行了研究。此外,点集的伪三角剖分亦是人们关注的。

三角剖分在许多领域内获得了良好的应用。例如,有限元分析、物体表面的表示、三维地形的描述、地下油层分布结构的描绘等都应用了三角剖分。

多边形的中轴

具有下述性质的点 q 的轨迹称为多边形 P 的中轴(或称骨架),多边形边界 δP 上存在两个(或两个以上)不同点 P_i 与 P_j 至 P 内点 q 距离相等。按此定义求得的中轴,其边不是直线段就是曲线段。由于出现曲线段,故给某些应用带来不便。周培德提出如下定义:多边形 P 的中轴为 P 内的点集,该点集中的点与多边形边界 δP 中两条或两条以上边(或边的延长线)距离相等。由此定义求得的中轴不再包含曲线段。如图 2 所示,无论由哪个定义求得的中轴,其结构形式均为一棵树。

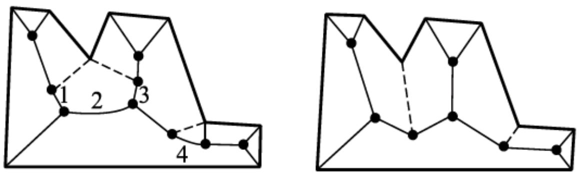


图 2　两种中轴定义的比较

在某种程度上,多边形 P 的中轴结构反映了多边形的形状特征,因而为了识别两个多边形是否相似,只要比较它们的中轴是否相似即可。中轴已在模式识别,计算机视觉等领域中得到应用,还可以利用中轴划分多边形成若干子多边形。

利用计算几何中的概念、方法可以设计出求解困难问题的有效算法。例如,货郎担问题。利用凸壳、中轴及点团的概念可以设计出求解货郎担问题的有效算法。实践证实,所求得的回路长度较其他方法(演化算法、蚁群算法、非齐次免疫算法、带聚类处理的蚁群算法等)求得的长度短。这样的例子不少,这表明计算几何是一门十分有用的学科,值得我们去学习、研究并加以推广。

参考文献

1. 周培德. 计算几何——算法设计、分析及应用. 5 版. 北京: 清华大学出版社, 2016

2. de Berg M, Cheong O, van Kreveld M, et al. 计算几何:算法与应用. 3 版. 邓俊辉,译. 北京: 清华大学出版社, 2009

3. 汪嘉业,王文平,屠长河,等. 计算几何及应用. 北京: 科学出版社, 2011

(周培德)

jisuan lilun

计算理论(theory of computation)　关于计算和计算机械的数学理论。

1936 年,为了讨论对于每个问题是否都有求解的算法,数理逻辑学家提出了几种不同的计算模型的定义。K. Gödel 和 S. C. Kleene 等人创立了递归函数论,将数论函数的算法可计算性刻画为递归可枚举性。A. M. Turing 和 E. L. Post 彼此独立地提出了理想计算机的概念,将问题的算法可解性刻画为在具有严格定义的理想计算机上的可解性。这一时期提出的通用图灵机在很大程度上影响了 1946 年出现的程序存储式计算机的设计思想。近代计算机的诞生使研究的焦点从理论可计算性转移到现实可计算性。因而,对一般算法设计方法的研究以及对一类问题的算法解的难度的分析与研究便成为计算机科学的热点。由此产生了算法学和计算复杂性理论等新兴的研究领域。自 A. M. Turing 提出图灵机, E. L. Post 提出波斯特机以后,直到 1948 年才由美籍匈牙利数学家 J. von Neumann 提出建立自动机的一般数学理论,对各种人造自动机和天然自动机进行比较、分析,探索其共同规律。他还研究了自动机的自繁殖和自恢复问题。1954 年前后形成了时序机的概念,1963 年后概率自动机的理论有了新的发展。1960 年后,细胞自动机在生物学和大规模集成电路技术的基础上有了重要的进展,成为自动机理论中的一个活跃领域。形式语言的研究始于 20 世纪初,把形式语言用于描述自然语言则是 20 世纪 50 年代中期的事。1956 年,N. Chomsky 发表了用形式语言方法研究自然语言的第一篇文章。在 1960 年发表的 ALGOL 60 报告中,首次使用与上下文无关文法等价的巴克斯范式系统来描述程序设计语言的局部语法,从而使形式语言理论在程序设计语言的设计与实现中发挥了重要的基础作用。

计算理论主要内容包括**算法**、**算法学**、**计算复杂性理论**、**可计算性理论**、**自动机理论**和**形式语言理

论等。

算法 解题过程的精确描述。它由有限条可完全机械执行的、有确定结果的指令(或命令、语句)构成并具有以下性质:①将算法作用于特定的输入集或问题描述上将产生由有限个动作构成的动作序列;②该动作序列具有唯一的初始动作;③除末尾的动作外,序列中的每个动作都有一个或多个后继动作;④序列或者终止于问题的解,或者终止于指出对给定输入集原问题无解。最常用的算法有**排序算法**,求解组合问题和组合最优化问题的**组合算法**等,而动作具有随机性的概率算法可望比确定型算法更高效。随着并行处理机的出现,一类适应于并行操作的**并行算法**便应运而生,并且日益成为热门的研究领域。

算法学 系统地研究算法的设计、分析与验证的学科。设计是创建算法的过程,算法学的任务之一就是研究一般的有效的算法设计方法。经典的方法有:**递归法**、**分治法**、**动态规划法**、**贪心法**、**回溯法**和**分支限界法**等。当然更多的问题还得具体情况具体分析。验证在于证明算法的正确性。基本途径是循环不变式加数学归纳法。分析是为了确定算法的效用或者说算法的复杂性以便比较求解同一问题的各种算法的优劣。

计算复杂性理论 用数学方法研究各类问题的计算复杂性的学科。算法复杂性是针对特定算法而言的,而计算复杂性则是针对特定问题而言的,后者反映的是问题的固有难度。计算复杂性等于最佳算法复杂性。最常用的**复杂性度量**有**时间复杂性**和**空间复杂性**。**复杂性归约**是比较问题的复杂性的重要工具。根据复杂性的阶可对被计算的问题分类。该学科的一个基本问题是:在计算时间多项式有界时,确定型机器与非确定型机器的解题能力是否相等?即P是否等于NP?

可计算性理论 研究计算的一般性质的数学理论。它通过建立计算的数学模型,精确区分哪些问题是可计算的,哪些问题是不可计算的。计算的过程就是执行算法的过程。根据图灵论题:图灵机可计算函数类与直观可计算函数类相同。**原始递归函数**是一类基本的直观**可计算函数**,但并非所有直观可计算的函数都是原始递归函数。然而,已经证明:部分递归函数类与图灵机可计算函数类相同。该学科的一个基本结论是:不可计算的函数要比可计算的函数多得多。特别地,虽然许多问题是可判定的,但更多的问题是不可判定的,如著名的**停机问题**和**波斯特对应问题**都是不可判定的。

自动机理论 以研究离散数字系统的功能和结构以及两者之关系为主要内容的数学理论。自动机是离散系统的抽象模型。给定自动机的功能描述,综合出具有这种功能的自动机的结构;给定自动机的结构,分析出它的功能,这就是自动机的综合与分析问题。自动机理论一般包括**有限自动机**理论、无限自动机理论、**概率自动机**理论和**细胞自动机**理论等等。

形式语言理论 用数学方法研究自然语言(如英语)和人工语言(如程序设计语言)的语法的理论。它只研究语言的组成规则,不研究语言的含义。该学科主要研究形式语言的描述工具、文法分类、语言分类以及各类语言的性质及其间的关系等。因为自动机可看成形式语言的一种识别器,所以形式语言理论与自动机理论有着十分密切的关系。

计算理论作为计算机科学的理论基础之一,其基本思想、概念和方法广泛应用于计算机科学的各个领域。早期的通用图灵机模型就是后来的程序存储式计算机模型的基本原型。递归函数的思想用于程序设计,产生了递归过程和递归数据结构,也影响了计算机的体系结构。可计算性理论的许多结论使人们不再为求解不可判定问题而浪费时间。算法学中提出的各种一般的算法设计方法广泛应用于各种具体问题的算法设计中,算法验证技术与程序验证技术互相借鉴。算法分析技术不但有助于判断给定算法是否现实可计算,而且有助于比较算法的质量。自动机作为一种基本工具已用于程序设计语言的编译程序中。线性自动机作为伪随机序列产生器用途广泛。可逆自动机用于通信编码。细胞自动机对并行计算机的设计思想产生了明显的影响。自繁殖自动机用于研究生物发育。形式语言理论在自然语言的理解和翻译、计算机语言的描述和编译、社会和自然现象的模拟、语法制导的模式识别等方面有广泛的应用。

参考文献

1. Aho A V, Hopcroft J E, Ullman J D. The design and analysis of computer algorithms. Boston, MA: Addison-Wesley, 1974

2. Papadimitriou C H. Computational complexity. Boston, MA: Addison-Wesley, 1994

3. Cutland N. Computability, an introduction to recursive function theory. Cambridge, UK: Cambridge University Press, 1980

4. Hopcroft J E, Ullman J D. Introduction to automata theory, languages, and computation. Boston, MA: Addison-Wesley, 1979　　(陈火旺　殷建平)

jisuan shulun

计算数论(computational number theory) 研究数论中计算方法的一门学科。其中两个最核心的问题是大整数的素性检验和因子分解。现代快速计算机的出现促进了计算数论的发展,只有利用先进的计算机,很多算法才可能实现。

计算数论与现代密码学有密切的关系。一个密码系统,对合法用户来说,它的加密、解密运算应该容易实现,但任何局外人要想破译它却是困难的。数论中就有很多计算问题,它同时具有“容易”和“困难”两个方面。例如,把两个大整数相乘是容易的,但在不知道这两个因子时,要将其乘积因子分解却是十分困难的。在 20 世纪 70 年代提出的 RSA 公钥密码正是利用了这一特性。

给定一个自然数 n,要判断它是不是素数,这就是素性检验。最简单的办法是用不超过$\sqrt{n}$的所有素数逐个去除 n,如果都除不尽,则 n 就是素数,否则 n 就是复合数。利用这个方法可找到 n 的因子。但当 n 很大时,这个方法的计算量太大,是不可行的。当 n 为素数时,若整数 a 与 n 互素,则

$$a^{n-1} \equiv 1 \pmod n \tag{1}$$

(费马小定理)。如果能够找到一个 a 使式(1)不成立,即可断定 n 是复合数。相反的结论并不能成立,即若式(1)对某个 a 成立,也不能断定 n 为素数。实际上,存在这样的复合数,使式(1)对一切与 n 互素的 a 都成立。这样的复合数称为 Carmichael 数。例如,$n=561=3\times11\times17$ 就是一个 Carmichael 数。费马小定理实际上只能给出 n 为复合数的确切判断,它不能直接提供一个素性检验的有效算法。不过后来找到的很多素性检验算法,却是从费马小定理引申发展起来的。

当式(1)成立时,称 n 为以 a 为基的伪素数。设 $n-1=2^k r$,其中 r 为奇数。当 n 为素数时,将式(1)两端开方,可得 $a^{(n-1)/2} \equiv \pm 1 \pmod n$。对任意 n,若

$$a^r \equiv 1 \pmod n$$

或存在 $i(0 \leqslant i < k)$,使

$$a^{2^i r} \equiv -1 \pmod n \tag{2}$$

则称 n 是以 a 为基的强伪素数。若存在一个 a,使式(2)不成立,n 就是复合数。当 n 为复合数时,在区间$(2,n-1)$内至少有 3/4 的数使式(2)不成立。从此区间内随机取 l 个数,对每个数检查式(2)是否成立(Rabin 检验)。若每个数都能使式(2)成立,则称 n 通过该检验。当 n 为复合数时,它能通过 Rabin 检验的概率为 4^{-l}。确切地说,这个算法实际上是概率型复合性检验。当 n 不能通过该检验时,n 就是复合数。当 n 通过该检验时,它就“很可能”是素数。

利用广义黎曼猜想可以证明,当 n 为复合数时,在区间$(2,[2(\log n)^2])$内一定有一个数 a 使式(2)不成立。因此逐个检查该区间内的数能否使式(2)成立,就是一个肯定型的素性检验。但广义黎曼猜想尚未证明。

如果能将 $n-1$ 完全分解为素因子乘积,则有如下的肯定型素性检验,它不依赖任何未证明的数学假设。若能找到一个正整数 $a(<n)$,使

$$\begin{cases} a^{n-1} \equiv 1 \pmod n \\ \text{对 } n-1 \text{ 的任一素因子 } p\text{,有 } a^{(n-1)/p} \not\equiv 1 \pmod n \end{cases} \tag{3}$$

则 n 是素数。这时由 a 的乘方可产生 $1,2,\cdots,n-1 \pmod n$,即 $1,2,\cdots,n-1$ 都与 n 互素。仅能将$n-1$部分地因子分解时,有如下的肯定型素性检验。设 s 是 $n-1$ 的因子,s 能完全地分解为素因子之积,且 $s>\sqrt{n}-1$,若能找到一个正整数 $a(<n)$,使

$$\begin{cases} a^{n-1} \equiv 1 \pmod n \\ \text{对 } s \text{ 的任一素因子 } p\text{,有}(a^{(n-1)/p}-1,n)=1 \end{cases} \tag{4}$$

则 n 是素数(Pocklington 检验)。当 $n+1$ 能完全分解时,有相应的素性检验(Lucas-Lehmer 检验)。当 $n+1$ 仅能部分地分解到一定程度时,也有相应的素性检验。当 $n-1$ 与 $n+1$ 同时能部分分解到一定程度时,则有一种混合型的素性检验。一般地说,要将 $n-1$ 或 $n+1$ 因子分解并非易事。但当 n 具有某些特殊形状,譬如,$n=2^{2^r}+1$ 为费马数,或 $n=2^p-1$(p 为素数)为梅森数时,上述检验是可行的。

目前所找到的最有效的素性检验是雅可比和检验与复乘检验,这两个方法都不依赖 $n\pm1$ 的因子分解。雅可比和检验利用代数整数环中与费马小定理类似的一个性质。复乘检验利用有限域 F_p 上的椭圆曲线。利用这两个方法,很容易检验一个几百位的整数是否是素数。

大整数因子分解的算法大体上可分为两个类型。

一个类型称为“同余式的组合”。设 n 是要分解的复合数,如果找到两个自然数 X,Y,使 $1\leqslant X,Y<n$,XY 与 n 互素,且

$$X^2\equiv Y^2 (\mathrm{mod}\ n) \tag{5}$$

由于 $n|(X-Y)(X+Y)$,当 $n\nmid X\pm Y$ 时,$(X-Y,n)$ 就是 n 的真因子,于是计算 $(X-Y,n)$ 就可将 n 分解。大体上说,出现 $n\nmid X\pm Y$ 的概率为 1/2,所以能找到形如式(5)的同余式越多,能分解 n 的可能性就越大。例如,若找出 10 个这样的同余式,就有 99.9%的可能性将 n 分解。问题归结为如何构造出一批形如式(5)的同余式。设 $p_1(=2),p_2,\cdots,p_k$ 为最小的 k 个素数,首先设法找出一批整数 $\{a_i|1\leqslant i\leqslant l\}$,使 $a_i^2\equiv b_i(\mathrm{mod}\ n)$ $(0<b_i<n)$,其中每个 b_i 的所有素因子都在 $p_1,p_2,\cdots,p_k$ 之中。然后在 $b_1,\cdots,b_l$ 中找出一些数 b_i,使它们的乘积为一个平方数。把相应那些 b_i 的同余式 $a_i^2\equiv b_i(\mathrm{mod}\ n)$ 相乘,便得到一个形如式(5)的同余式。

D. H. Lehmer 和 R. E. Powers 于 1931 年提出连分数因子分解算法,利用$\sqrt{n}$的连分数展开式找到$\sqrt{n}$的一批最佳渐近分数a_i/c_i,设 $a_i^2\equiv b_i(\mathrm{mod}\ n)$,$0<b_i<n$。连分数理论指出 $b_i\leqslant 2\sqrt{n}$,由于 b_i 较小,就有较大的可能找出一批 b_i,使其素因子都在最小的 k 个素数之中。称最小的 k 个素数的集合为因子基,也可以选择其他素数集合为因子基。通过因子基寻找同余式(5)的系统方法,是在 20 世纪 70 年代初由 M. Morrison 和 J. Brillhart 提出的。M. Kraitchik 于 1926 年提出利用二次函数 $f(x)=(x+[\sqrt{n}])^2-n$。由于 $(x+[\sqrt{n}])^2\equiv f(x)\ (\mathrm{mod}\ n)$,$f(x)$ 相当于上述 b_i,可以使它较小。C. Powerance 于 1982 年提出了一个系统的方法,对于相连的一串整数 x,挑出那些素因子都在预先选定的因子基内的 $f(x)$。这个因子分解的算法称为二次筛法,这是目前一个有效的因子分解算法。它对于被分解的整数的形式没有特殊要求,曾用它分解 111 位的整数。它的计算量大约为$\exp((1+0(1))(\lg n\ \lg\lg n)^{1/2})$。

大整数因子分解的另一类型算法是都利用一个群,且这个群的阶(即元素个数)不含大素因子。设 p 是 n 的素因子,$M(k)$ 为最小的 k 个自然数的最小公倍数。若 $p-1|M(k)$,则 $2^{M(k)}\equiv 1\ (\mathrm{mod}\ p)$(费马小定理)。如果 $2^{M(k)}\not\equiv 1\ (\mathrm{mod}\ n)$,则 $(2^{M(k)-1},n)$ 就是 n 的真因子,从而 n 可以分解。对于一批较小的 k,计算 $(2^{M(k)-1},n)$,希望能找到 n 的一个真因子(Pollard 算法,1974 年提出)。这里利用了群 F_p^*(含 p 个元素的域的乘法群),当 $|F_p^*|=p-1$ 的素因子都较小时,就有可能分解 n。显然,也可以利用其他的群。H. W. Lenstra 于 1987 年提出椭圆曲线因子分解算法,利用了 F_p 上的椭圆曲线的点所成的群,当该群的阶适合上述要求时,就有可能将 n 分解。对于固定的一个 p,有很多条 F_p 上的椭圆曲线可供选择使用,从而增加了分解 n 的机会。

在强化二次筛法和椭圆曲线算法的优点的基础上,John Pollard 于 1988 年提出数域筛法。1990 年分布在世界各地的很多学者通过计算机联网,在数个月内,利用数域筛法共同分解了第 9 个费马数 $2^{2^9}+1$,它是一个 7 位素数、一个 49 位素数和一个 99 位素数的乘积,这是一个特殊形式的整数。

参考文献

1. Pomerance C. Cryptology and computational number theory. American Mathematical Society, 1990
2. Koblitz N. A course in number theory and cryptography. New York: Springer-Verlag, 1987
3. Cohen H, Lenstra A K. Primality testing and Jacobi sums. Math. Comp., 1984, 48: 103-121
4. Goldwasser S, Kilian J. Almost all primes can be quickly certified. Proc. 18th Annual ACM Symp. on Theory of Computing, 1986: 316-329
5. Lenstra H W. Factoring integers with elliptic curves. Ann. of Math., 1987, (126): 649-673

(裴定一)

jisuan xuexi lilun

计算学习理论(computational learning theory) 试图阐明什么样的问题是可学习的,可学习问题的学习效率如何,从而为设计机器学习算法提供理论依据。

计算学习理论中最重要的是**概率近似正确**(probably approximately correct)**学习模型**,简称 PAC 学习,由 Valiant 提出。PAC 学习要求给定随机抽取的训练数据后,算法能以很高的概率从函数集中学出一个与最优函数误差很小的函数。如果存在一个学习算法使得对任意 $0<\delta<1/2,\varepsilon>0$,算法能以至少 $1-\delta$ 的概率输出一个 ε-近似正确的结果,并且所需要的训练数据个数随 $1/\varepsilon$ 和 $1/\delta$ 至多以多项式阶增长,则称 PAC 可学习。换言之,PAC 可学习意味着用有限的训练样本即可学出较为可靠的结果。

计算学习理论研究各种学习问题是否 PAC 可学习。重点在于刻画 PAC 可学习的条件。这方面最重要的理论结果是 VC 维，由 Vapnik 与 Chervonenkis 得到：若函数集的 VC 维数是有限值，则无论数据的概率分布如何都是 PAC 可学习的。

最初的 PAC 学习模型中假定数据不含噪声，这一假设在实际应用中很难满足。不可知(agnostic) PAC 模型对 PAC 进行了改进，允许数据含任意(不为算法所知的)噪声。在此模型下，VC 维数仍是可学习的充分必要条件。

对 PAC 模型的另一个推广是允许学习精度有所放松。PAC 中要求精度参数 ε 可以任意小，这称作强可学习。若仅要求精度比随机猜测的结果略优，则称为弱可学习。计算学习理论中一个非常重要的结果是弱可学习等价于强可学习。通过一种称作 boosting 的技术，可以将一个弱学习算法推进为强学习算法。

参考文献

1. Valiant L. A theory of the learnable. Communications of ACM, 1984, 27(11): 1134-1142

2. Kearns M, Vazirani U. An introduction to computational learning theory, MIT Press, 1994

(王立威)

jisuan yuyanxue

计算语言学(computational linguistics)　通过建立形式化的计算模型来分析、理解和生成自然语言的学科。它是**人工智能**和语言学的分支学科。计算语言学是典型的交叉学科，其研究常常涉及计算机科学、语言学、数学等多个学科的知识。与内容接近的学科**自然语言处理**相比较，计算语言学更加侧重基础理论和方法的研究。

计算语言学研究大致始于 20 世纪 50 年代，初期的目标是因应**机器翻译**研究对基础理论和方法的需求，后来逐渐发展成一个体系自洽的独立学科，服务的应用领域也不再限于机器翻译。

计算语言学的主要研究内容包括：①建立形式化的适于计算机处理的语言模型。语言模型可以表现为符号化的规则，也可以表现为数值化的数据，如词法、句法和语义等知识的形式描述体系、不同层级语言对象的形式表示方法、面向计算的形式化语法理论以及揭示语言现象统计分布的概率模型。②基于所建立的符号化或数值化模型，研制分析、理解和生成语言的各种技术和算法，如面向语言理解的词语切分、词性标注、形态分析、句法分析、语义分析、指代消解等技术和算法，以及面向**自然语言生成**的文档规划、指称表达的生成、词汇选择和表层实现等技术和算法。与早期相比，计算语言学研究在上述两个方面都不乏创新和进展。但计算语言学的相关研究仍然充满挑战，这主要是因为自然语言与人工语言不同，没有完备确定的规则系统，歧义现象严重，其理解和生成不仅需要语言本身规律的支撑，也离不开常识和各种百科专业知识，这些知识量大、形式化困难，目前人们还缺乏可靠有效的手段。

从研究方法层面看，计算语言学在其发展早期就形成了两种并行不悖的研究路线，即基于规则的理性主义路线和基于统计的经验主义路线。20 世纪 90 年代以前，由于缺乏大规模的电子化语言样本资源，且受机器计算性能所限，人们更多采用的是基于规则的理性路线。20 世纪 90 年代以后，随着各种大规模语言资源的相继建成以及计算机性能的持续提升，各种统计技术和**机器学习**技术在计算语言学中也得到了密集使用，并取得显著进展。

计算语言学研究有着广泛的应用领域，其研究成果可直接服务于机器翻译、智能人机接口、信息检索与提取、文本摘要与分类等实际问题的解决，具有很大的实际应用意义和经济价值。60 多年来，各国研究人员努力建造了大量各类语言资源，积累了大量基础模型和方法，在这些资源和方法的支持下，各类应用系统的性能和水平也持续得到提升。不过，要实现机器理解自然语言和生成自然语言这个终极目标，可能还需要比较漫长的研究才有可能。

参考文献

1. 俞士汶. 计算语言学概论. 北京：商务印书馆, 2003

2. Jurafsky D, Martin J H. Speech and language processing: An introduction to natural language processing, computational linguistics and speech recognition. 2nd ed. Upper Saddle River, NJ: Prentice Hall, 2008

(常宝宝)

jisuan yuyanxue yufa lilun

计算语言学语法理论(grammatical theory of computational linguistics)　关于自然语言语法知识的系统的形式化论述。可以分为基于短语结构的语法和基于词间关系的语法两大类。前者认为句子由短语组成，短语由词组成，语法就是在区分短语和词的不同范畴基础上描述范畴间的组合规

则,代表性的语法理论有上下文无关语法;后者认为句子由词直接组成,无须经过短语这个中间层次,语法就是描述词与词之间的支配与依存关系,代表性的语法理论有依存语法。这两大类语法理论都可以通过树图形式来呈现句子的结构,但具体的呈现方式有所不同。以"张三丢了一双鞋"为例,基于上下文无关语法对句子结构的描述如图1所示;基于依存语法对句子结构的描述如图2所示。

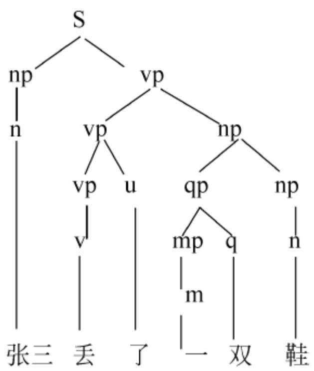


图 1

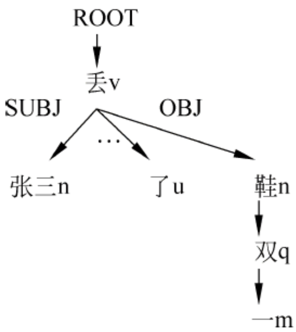


图 2

图1中S代表句子,np,vp等分别代表名词性短语,动词性短语等短语范畴。n、v等分别代表名词、动词等词类范畴。句子的树状结构规定了词语间的组合顺序,以及每个组合的内部语法关系,比如组成S的np与vp之间是主谓结构关系。图2中ROOT代表句子的"根"(或中心),由谓语动词"丢"充当,动词支配主语"张三"、时态成分"了"、宾语"鞋"。句子的树状结构规定了句中词语间的支配与依存关系,箭头上方的节点是支配词语,下方的节点是依存词语。箭头线上还可以进一步标记词语间依存关系的不同类型。比如"SUBJ"表示主语-动词依存关系,"OBJ"表示宾语-动词依存关系,等等。

由于自然语言单位组合的复杂多样性,在计算语言学理论研究和**自然语言处理**的实践过程中,基于上述两大类基础语法理论框架,发展出了许多侧重点不同的语法理论模型。其中有代表性的语法理论模型包括:将短语结构区分为深层结构和表层结构来揭示不同句法结构之间关联的转换生成语法;引入特征结构与合一约束等形式表达机制的词汇功能语法、功能合一语法;强调短语结构内部中心成分作用的中心语驱动短语结构语法、广义短语结构语法、X-bar语法;可以直接用Prolog程序语言来书写规则的限定子句语法;以树结构的替换和拼接操作代替短语两两组合方式来描述语言单位组合规律的树邻接语法;在短语结构语法基础上增加规则概率信息的概率上下文无关语法;以约分形式表达词范畴的范畴语法;以算子形式表达词范畴的链语法;侧重描述动词与名词组配关系的格语法、配价语法;侧重描述从表达功能到语法形式对应关系的系统功能语法,等等。

参考文献

1. Mitkov R, ed. The Oxford handbook of computational linguistics. Oxford University Press, 2003

2. Baltin M, Collins C. The handbook of contemporary syntactic theory. Blackwell Publishers Ltd. 2000

3. 俞士汶. 计算语言学概论. 北京:商务印书馆,2003

(詹卫东)

jisuan zhineng

计算智能(computational intelligence) 以数据处理为基础的一类问题求解的智能方法和技术。计算智能是受到大自然智慧和人类智能的启发,模仿生物界的进化过程,或模仿生物的生理构造和身体机能,或模仿动物的群体行为,或模仿人类的思维、语言和记忆过程的特性,或模仿自然界的物理现象,希望通过模拟大自然和人类的智慧实现对问题的优化求解,在可接受的时间内求出可以接受的解。这些求解的算法共同组成了计算智能优化算法。

作为**人工智能**的一个重要领域,计算智能因其智能性、并行性和健壮性,具有很好的自适应能力和很强的全局搜索能力,得到了众多研究者的广泛关注,目前已经在算法理论和算法性能方面取得了很多重要的进展。计算智能领域典型的方法和算法有**人工神经网络**、**模糊逻辑**、**遗传算法**、蚁群优化算法、粒子群优化算法、免疫算法、分布估计算法、模因(Memetic)算法、模拟退火算法和禁忌搜索算法、人工生命等。这些方法具有以下共同的要素:自适应的结构、随机产生的或指定的初始状态、适应度的评

测函数、修改结构的操作、系统状态存储器、终止计算的条件、指示结果的方法、控制过程的参数。计算智能的这些方法具有自学习、自组织、自适应的特征和简单、通用、鲁棒性强、适于并行处理的优点。

目前,计算智能算法在国内外得到广泛的关注,已经成为人工智能的重要研究方向。计算智能还处于不断发展和完善的过程,目前还没有牢固的数学基础,国内外众多研究者也是在不断的探索中前进。计算智能技术在自身性能的提高和应用范围的拓展中不断完善。计算智能的研究、发展与应用,无论是研究队伍的规模、发表的论文数量,还是网上的信息资源,发展速度都很快,已经得到了国际学术界的广泛认可,在优化计算、**模式识别**、图像处理、自动控制、经济管理、机械工程、电气工程、通信网络和生物医学等多个学科领域取得了成功的应用,涉及国防、科技、经济、工业和农业等各个方面。

参考文献

Eberhart R C, Yuhui Shi. Computational intelligence: Concepts to implementations. Morgan Kaufmann Publishers, 2007

(史忠植)

jilu leixing

记录类型(record type)　分量可以具有不同类型的一种构造类型。记录是收集相关成分的一种**数据结构**。例如,关于人(person)的记录可包括名字(name)、性别(sex)及年龄(age)等,用 PASCAL 语言可将此记录类型定义如下:

```
type
    person  = record
                    name:array[1..10] of char;
                    sex:(male,female);
                    age:1..100
              end;
var
    chen,wang:person;
```

这里 name,sex,age 为域名,用来表记记录中的不同分量,每个域名还有一个定义其值范围的类型。person 为定义该记录的类型的标识符,通过它可创建诸如 chen,wang 等不同的记录实例。

记录中允许域的数据类型在表达形式上有所变化。例如,空间中点的位置可按直角坐标(x,y,z)或按极坐标(r,φ,z)的形式给定,因此在“点”类型的记录中,允许出现变化的数据形式。PASCAL 语言允许构造这种**变体记录**:

```
type
    coordinate = (rectangular,cylindrical);
    point = record
            z:real;
    case c:cordinate of

    rectangular:(x,y:real);

    cylindrical:(r,phi:real)
end;
```

这里,变体选择符 c 描述了实际出现的可选情况。

记录类型的操作除了赋值和相等比较这两种操作外,还有记录的构成与选择,即从各分量的值来构造出的记录类型的一个值,和从记录类型的一个值中选择出其组成成分。　(陈涵生)

jicheng

继承(inheritance)　基于层次关系的不同类间共享数据和操作的关系。通过继承,一个类的定义可以基于另一个类(后者称作前者的基类):对基类的特征或是不加任何修改,或是根据特定情形作适当调整。

作用　继承是面向对象软件开发的重要设施,它有效地支持软件复用与扩充。类既是模块,又是类型。对应于类的这两种作用,继承可作为:①模块扩充机制,它能通过增加或修改已有类的特征来定义新类;②类型精化机制,它支持通过例化已有类型来定义新类型。

种类　根据继承关系的性质,可区分如下几种继承:

(1)直接继承和间接继承:如果类 C 的定义直接使用类 B 中的特征,则称 C 直接继承 B,且称 B 为 C 的直接基类,C 为 B 的直接衍类。如果类 C 直接继承类 B,类 B 直接继承类 A,则称 C 间接继承 A,且称 C 为 A 的间接衍类,A 为 C 的间接基类。间接继承体现了继承关系的可传递性。

(2)单继承和多继承:如果一个类只有一个直接基类,则称该继承关系为单继承;如果一个类有多个直接基类,则称该继承关系为多继承。

(3)重复继承:若类 D 的多个直接基类有共同基类 A 时,则称 D 重复继承 A,称 A 是 D 的重复基类。

参考文献

1. 徐家福,等. 对象式程序设计语言. 南京:南

京大学出版社,1992

2. Special Issue on Object-Oriented Design. CACM,1990,33(9) (张家重 王志坚)

jiafaqi

加法器(adder) 对以数字形式表示的两个 n 位数求和的一种运算电路。加法器是计算机中最基本的运算部件。

基本的加法单元称为全加器,它要求三个输入量,即操作数 X_n、Y_n 和低位传来的进位 C_{n-1},并产生两个输出量,即本位和 F_n 及向高位的进位 C_n,F_n 和 C_n 的表达式为

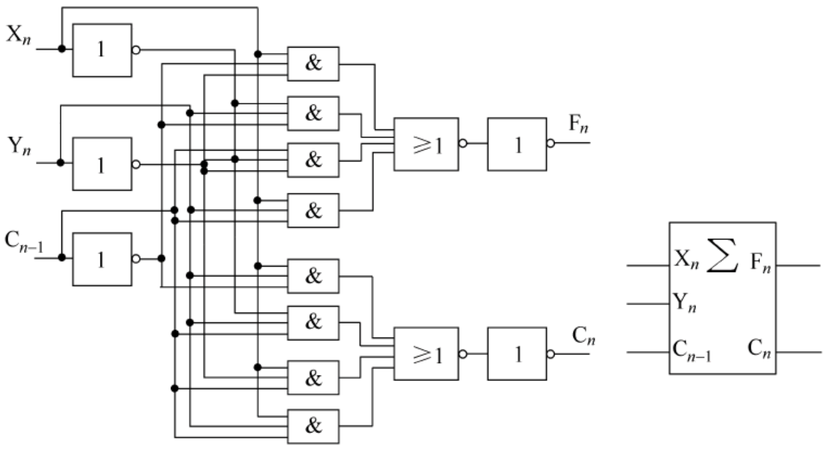
$$F_n = X_n\bar{Y}_n\bar{C}_{n-1} + \bar{X}_nY_n\bar{C}_{n-1} + \bar{X}_n\bar{Y}_nC_{n-1} + X_nY_nC_{n-1}$$

$$C_n = X_nY_n\bar{C}_{n-1} + X_n\bar{Y}_nC_{n-1} + \bar{X}_nY_nC_{n-1} + X_nY_nC_{n-1}$$

全加器的功能表见图1(a),逻辑图见图1(b),逻辑符号图见图1(c)。

X_n	Y_n	C_{n-1}	F_n	C_n
0	0	0	0	0
0	0	1	1	0
1	0	0	1	0
1	0	1	0	1
0	1	0	1	0
0	1	1	0	1
1	1	0	0	1
1	1	1	1	1

(a) 功能表



(b) 逻辑图 (c) 逻辑符号图

图1 全加器的功能表、逻辑图及逻辑符号图

n 位加法器可分为串行加法器和并行加法器。在串行加法器中,只用一位全加器,数据逐位串行进入加法器进行运算。串行加法器具有器件少、成本低的优点,但运算速度太慢,在计算机中几乎不使用。并行加法器的位数和操作位数相同,能够在一步操作中对各位同时相加,影响加法时间的主要因素是进位信号的传递时间。

按进位链来分,并行加法器可分为串行进位和并行进位两种。

并行加法器的每一位全加器都有一个从较低位送来的进位和一个向较高位的进位,将各位之间的进位线路连接起来,构成进位链。每一位的进位表达式为

$$C_i = A_i \cdot B_i + (A_i \oplus B_i) \cdot C_{i-1}$$

其中:$A_i \cdot B_i$ 取决于本位参加运算的两个数,而与低位进位无关,因此称 $A_i \cdot B_i$ 为进位产生函数(本位进位产生),用 G_i 表示;$(A_i \oplus B_i) \cdot C_{i-1}$ 则不但与本位的两个数有关,还依赖于低位送来的进位,因此称 $A_i \oplus B_i$ 为进位传递函数(低位进位传递),用 P_i 表示。

将 n 个全加器相连可得 n 位串行进位加法器(图2),其加法时间较长。这是因为其位间进位是串行传递的(也称之为行波进位),本位全加和 F_i 的操作必须等低位进位 C_{i-1} 到来后才能进行,加法时间与位数有关。并行进位又叫先行进位或超前进位,特点是各级进位信号同时形成,其值如下

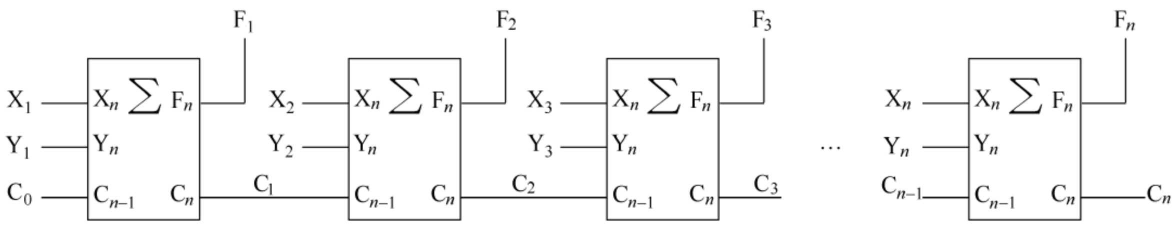


图2 串行进位加法器

$$C_1 = G_1 + P_1 \cdot C_0$$
$$C_2 = G_2 + P_2 \cdot C_1 = G_2 + P_2 \cdot G_1 + P_2 \cdot P_1 \cdot C_0$$
$$C_3 = G_3 + P_3 \cdot C_2 = G_3 + P_3 \cdot G_2 + P_3 \cdot P_2 \cdot G_1 + P_3 \cdot P_2 \cdot P_1 \cdot C_0$$
$$C_4 = G_4 + P_4 \cdot C_3 = G_4 + P_4 \cdot G_3 + P_4 \cdot P_3 \cdot G_2 + P_4 \cdot P_3 \cdot P_2 \cdot G_1 + P_4 \cdot P_3 \cdot P_2 \cdot P_1 \cdot C_0$$
$$\vdots$$

所有各位的进位都直接从 C_0 并行产生,这种进位方式速度快,但是随着位数的增加,C_i 的逻辑表达式变得越来越长,这会使电路结构变得很复杂。通常采用分组并行进位方式,它是把 n 位字长分为若干小组,在组内各位之间实行并行快速进位,而在组间可以采用串行进位方式,或是采用并行快速进位方式。

美国 Intel 公司的 74283 芯片就是一个 4 位超前进位加法器,利用 4 片 74283 就可组成一个 4 位一组、组内并行进位、组间串行进位的 16 位加法器。

参考文献

1. 蒋本珊. 电子计算机组成原理. 北京:北京理工大学出版社,1994

2. 王爱英. 计算机组成与结构. 3 版. 北京:清华大学出版社,2001

(刘恩德)

jiagu jishu

加固技术(ruggedization technology) 为使产品适应恶劣环境或确保其使用安全而采取的防护技术。加固技术主要包括抗振动、抗冲击技术,热设计技术,电磁兼容设计技术,抗核爆炸技术及三防(防潮湿、防沙尘、防烟雾和霉菌)技术和**防信息泄露技术**等。

常用的加固方法有先天加固与后天加固。

先天加固又称为内加固,指按照加固要求进行元器件选择,电源、电路板及机箱设计,从而使加固后的产品能满足恶劣环境要求。先天加固的优点是加固性能好,可适用于最恶劣的环境。缺点是价格昂贵,技术、工艺复杂,对电路及元器件的要求高,研制周期长。

后天加固又称为外加固,是利用优选的民用产品,采用重新设计机箱以及某些能达到高温、低温、潮湿、冲击、振动要求的措施,从而满足初级加固环境要求。后天加固的优点是成本低,研制周期短,缺点是不能用于较为恶劣的环境。

根据加固程度和使用环境的不同,加固型产品可分为军用规范型(简称 M 型)、加固型(简称 R 型)、防信息泄露型(简称 T 型)三种类型。每一类型又可分为几个级别,如地面固定、车载、舰载、有(无)空调、野外等。军用规范型是严格按照一系列军用标准设计和制造的,可满足最恶劣的外部环境要求。加固型一般是采用后天加固的方法,使之满足一般恶劣环境的要求。防信息泄露型产品是指满足计算机信息泄露要求的专用型产品。防信息泄露型产品不一定是军用规范型或加固型产品。

(於亮)

jiasubi xingneng moxing

加速比性能模型(performance model of speed-up ratio) 指给定的一个性能模型,用以衡量对于一个给定的应用,采用并行方法(并行结构或并行算法)的执行速度相对于串行方法的执行速度加快了多少倍。依据不同的出发点,加速比可分为绝对加速比和相对加速比。绝对加速比侧重于比较并行处理与串行处理的性能和效果,通常以最优串行算法为基准。根据与具体的机器结构是否相关,绝对加速比又分为两种不同的定义:一种是与具体机器结构有关,即将最优串行算法在一台处理器上的运行时间与并行算法在多个处理器上的运行时间之比;另一种是与具体机器结构无关,即将最优串行算法在一台最快的顺序机上的运行时间与并行算法在并行处理机上的运行时间之比。相对加速比定义为同一并行算法在单处理器上的运行时间与在由多个相同单处理器构成的并行处理机系统上的运行时间之比,侧重于描述并行算法和并行计算机本身的可扩展性。依据性能模型不同,加速比又可分为固定负载加速比性能模型、固定时间加速比性能模型和固定存储容量加速比性能模型。

1. 固定负载加速比性能模型

该模型于 1967 年由 Gene M Amdahl 加速定律,又称 **Amdahl 加速定律**。其基本思想是在对实时性要求严格的应用中强调运行时间。假设计算负载固定,随着并行计算机中处理器数目的增加,每个处理器执行的平均工作负载减少,能够在更短的时间得出运行结果。

设 W 是工作负载,它可以定义为给定问题的总计算量,也称为工作负载、计算负载、问题规模等。W_s 表示工作负载中的不可并行化的工作量,W_p 表

示工作负载中的可并行化的工作量，满足 $W_s + W_p = W$。假设 N 是并行方法中的并行度（如采用 N 个处理器）；那么在并行度为 N 的并行处理机系统中，Amdahl 加速定律表明加速比可以表示为

$$S = \frac{W_s + W_p}{W_s + W_p/N}$$

若 f 代表问题中不可并行化负载所占的比例，即

$$f = W_s/W = W_s/(W_s + W_p)$$

那么并行化负载所占的比例是

$$1 - f = W_p/W = W_p/(W_s + W_p)$$

代入加速比公式即得 Amdahl 加速定律的表述公式为

$$S = \frac{1}{f + (1 - f)/N}$$

如果 $f = 0$，则得到理想的最大加速比为 $S = N$；随着 f 的增加，加速比性能下降；当 $f = 1$ 时，即串行负载占 100%（没有可以并行化的负载），则得到最小加速比为 $S = 1$（没有加速）；如果 $N \to \infty$，则得到极限加速比为 $s = 1/f$，说明随着并行度 N 的不断增大，系统加速比不断增加，但加速比的上限受限于不可并行化部分的负载比例。这表明，只要存在不可并行化的负载，就不能通过增加处理机数目 N 的方法来完全解决，虽然这一特性在过去给人们造成了并行处理能力潜力有限的悲观印象，然而 Amdahl 加速定律并没有否定并行计算的价值，相反，它揭示了要想充分发挥并行性能就必须考虑工作负载的组成特性。

2. 固定时间加速比性能模型

1988 年，John L Gustafson 提出了固定时间加速比模型，又称 **Gustafson 定律**。强调运算的精度而不是运算时间。求解问题的规模随并行计算机的规模增加，从而在运算时间保持不变的前提下得到更精确的解。问题规模的增加使得增加处理机后的所有处理机依然保持忙碌的工作状态，在问题规模和并行度都增加 N 倍的情况下，Gustafson 定律的加速比公式可以表示为

$$S = \frac{W_s + NW_p}{W_s + NW_p/N} = \frac{W_s + NW_p}{W_s + W_p}$$
$$= f + N(1 - f) = N - f(N - 1)$$

随着问题规模的不断增加，不可并行化部分所占比例 f 已经不是一个常数，而是越来越小，即 Gustafson 定律作为固定时间加速比模型，其加速比不再受限于不可并行化部分的负载。

3. 固定存储容量加速比性能模型

1990 年，Xian-He Sun 和 Lionel M Ni 提出了固定存储容量加速比性能模型，又称 **Sun 和 Ni 定律**。其基本思想是在计算机存储空间有限的条件下求解规模尽可能大的问题。现代的很多大型科学计算和工程设计应用问题需要较大的存储空间，而存储器受限需要通过扩展工作负载的方式为并行计算机提供较高的加速比、精度和资源利用率。Sun 和 Ni 定律的加速比公式表示为

$$S = \frac{W_s + G(N)W_p}{W_s + G(N)W_p/N}$$

$G(N)$ 代表当并行计算机的存储容量增加 N 倍时并行工作负载增加的倍数。当 $G(N) = 1$ 时，简化为固定负载加速比模型的情况，即 Amdahl 加速定律。当 $G(N) = N$，即存储容量增加 N 倍，工作负载也增加 N 倍时，成为固定时间加速比模型，即 Gustafson 定律。当 $G(N) > N$ 时，工作负载的增加比存储器容量的增加要快，此时系统的加速比要快于前两种情况。（李克秋）

jiating wangluo

家庭网络（home network） 将家用电子产品与常规家用电器相互连接，使得用户能够远程接入与控制这些产品与系统并获得相应数据的网络。这一定义是由美国电子消费品制造商协会（Consumer Electronics Association，CEA）的家庭网络与 IT 部门（Home Networking and IT，HNIT）提出的。家庭网络是一种日常生活的**局域网**，用于提供布置在家中的数字设备与电气设备之间的通信，数字设备通常包括少量的个人计算机和附属设备，例如打印机以及移动计算设备，电气设备则包括常用家用电器，如洗衣机、电冰箱、空调等。它的重要的功能之一是家用数字设备和电器设备的远程接入与统一控制，另外一个则是互联网接入的共享。

家庭网络既可以使用有线技术，也可以使用无线技术；有线系统通常使用双绞线，但也可以使用同轴电缆或现有的电线。无线技术也是一种普遍的构建家庭网络的方法，大部分产品都可以使用无线网络，协议包括 IEEE 802.11a、802.11b、802.11g 和 802.11n。除此之外红外与蓝牙技术在家庭网络领域也有非常广泛的应用。

参考文献

Rose B. Home networks：a standards perspective. IEEE Communications Magazine，2001，39(12)：78-85

（崔勇）

jiashe jianyan

假设检验(hypothesis testing) 构造一个统计量,根据给定的子样计算它的值,从而判断有关母体分布假设是否正确的统计推断方法。当母体分布函数形式未知时,如要检验母体分布是否为某种形式的分布,则称为非参数假设检验;当母体分布形式已知,但其中有未知参数,如检验这些参数是否满足某种假设,则称为参数假设检验。下面主要叙述参数假设检验。设母体分布为 $F(x;\theta)$,θ 为未知参数,其取值空间为 Θ。首先要提出原假设 $H_0:\theta\in\Theta_0$;对立假设 $H_1:\theta\in\Theta_1$(Θ_0,Θ_1 均为 Θ 的子集)。假设检验问题就是要构造一个统计量 $T=T(X_1,X_2,\cdots,X_n)$,根据子样的取值$(x_1,x_2,\cdots,x_n)$,计算 $T(x_1,x_2,\cdots,x_n)$,并根据它的值来确定子样空间 $\mathscr{X}$ 的一个划分;$\mathscr{X}=\mathscr{X}_0^*\cup\mathscr{X}_1^*,\mathscr{X}_0^*\cap\mathscr{X}_1^*=\varnothing$,其中 $\mathscr{X}_0^*$ 称为否定域,$\mathscr{X}_1^*$ 称为接受域。当$(x_1,x_2,\cdots,x_n)\in\mathscr{X}_0^*$ 时否定 H_0,否则接受 H_0。这里划分的根据是:当 $\theta\in\Theta_0$ 时,$p_\theta(\mathscr{X}_0^*)\triangleq p_\theta((X_1,X_2,\cdots,X_n)\in\mathscr{X}_0^*)$ 为一个很小的数 α,通常取$\alpha=0.05,0.01$ 等。根据小概率事件原理:“小概率事件在一次试验中是几乎不可能发生的”,既然$p_\theta(\mathscr{X}_0^*)=\alpha$,而今居然观察到$(x_1,x_2,\cdots,x_n)\in\mathscr{X}_0^*$,所以有理由认为 θ 不属于 Θ_0,也即否定 H_0。但这种判断有可能错误,称 $\alpha=p_\theta(\mathscr{X}_0^*)$ 为犯第一类错误(弃真)的概率。同时,当 $\theta\in\Theta_1$,即 H_0 不真时,也希望判断错误的概率 $p_\theta(\mathscr{X}_1^*)$ 尽可能地小,称它为犯第二类错误(存伪)的概率,记为 β。在样本容量 n 固定时,不可能使 α 与 β 同时任意地小,一般只能在满足对 α 要求的条件下,寻求使势函数 $p_\theta(\mathscr{X}_0^*)=1-\beta(\theta\in\Theta_1)$ 尽可能地大。特别要指出的是,因为 α 很小,所以当原假设 H_0 为真时,否定 H_0 的概率很小,实际上起着保护原假设的作用,因此在应用中,总是把按过去的经验很可能为真的或者特别要注意排除的,取为原假设。称 α 达到要求的,势函数达到最大的检验为最佳检验。奈曼-皮尔逊引理指出:设母体分布密度为 $f(x;\theta)$,θ 未知,检验问题 $H_0:\theta=\theta_0,H_1:\theta=\theta_1$,则对任一常数 α $(0<\alpha<1)$,存在显著水平(即犯第一类错误概率)为 α 的最佳检验法,它的拒绝区域 $\mathscr{X}_0^*$ 由下式确定:

$$l(x_1,x_2,\cdots,x_n)=\frac{\prod_{i=1}^{n}f(x_i;\theta_1)}{\prod_{i=1}^{n}f(x_i;\theta_0)}\geqslant k \qquad (1)$$

其中 k 为满足 $p_{\theta_0}(l(x_1,x_2,\cdots,x_n)>k)=\alpha$ 的数。在奈曼-皮尔逊引理中,Θ_0,Θ_1 均为单点集,这种检验称为简单假设检验,否则称为复合假设检验。当 H_1 为复合假设时,不一定存在最佳检验法。可以证明,当 H_1 是单边的情形,比如 $\theta<\theta_1$ 或 $\theta>\theta_1$,则存在一致最佳检验(即对一切 $\theta>\theta_1$,势函数$p_\theta(\mathscr{X}_0^*)$都达到最大)。

最常用的参数检验法是 u 检验法、t 检验法与 χ^2 检验法。u,t 检验法都是用来检验正态母体中关于均值的假设,前者适用于方差已知的情形,后者用于方差未知的情形。χ^2 检验则用来检验正态母体关于方差的假设。

(1)设有正态母体 $N(\mu,\sigma_0^2)$,σ_0^2 已知,检验 H_0:$\mu=\mu_0,H_1:\mu\neq\mu_0$。可以证明当 H_0 成立时,统计量 $(\overline{X}-\mu_0)\Big/\frac{\sigma}{\sqrt{n}}$ 服从 $N(0,1)$ 分布,因此可得拒绝区域为:$|\overline{X}-\mu_0|>N_{\alpha/2}\,\sigma/\sqrt{n}$,其中 $N_{\alpha/2}$ 为正态分布的α临界值,即正态$N(0,1)$变量X大于$N_{\alpha/2}$的概率为$\alpha/2$。由此可得关于μ的置信度为$1-\alpha$的置信区间为

$$(\overline{X}-N_{\alpha/2}\,\sigma/\sqrt{n},\quad \overline{X}+N_{\alpha/2}\,\sigma/\sqrt{n}) \qquad (2)$$

(2)上述问题中,若 σ^2 未知,则可证明当 H_0 成立时,统计量 $T=\sqrt{n}(\overline{X}-\mu_0)/s_0$ 服从 t_{n-1} 分布,其中 s_0^2 为子样的修正方差 $s_0^2=\frac{1}{n-1}\sum_{i=1}^{n}(X_i-\overline{X})^2$。于是可得拒绝域为:$|\overline{X}-\mu_0|>(s_0/\sqrt{n})t_{n-1}(\alpha/2)$,其中 $t_{n-1}(\alpha)$为 t_{n-1} 分布的临界值,可从 t 分布表查得。由此可得关于 μ 的置信度为 $1-\alpha$ 的置信区间是

$$(\overline{X}-t_{n-1}(\alpha/2)s_0/\sqrt{n},\overline{X}+t_{n-1}(\alpha/2)s_0/\sqrt{n}) \qquad (3)$$

(3)设正态母体 $N(\mu,\sigma^2)$ 中 μ 已知,检验 $H_0:\sigma^2=\sigma_0^2$,$H_1:\sigma^2\neq\sigma_0^2$。在 H_0 为真时,$k=(1/\sigma_0^2)\sum_{i=1}^{n}(X_i-\mu)^2$ 服从χ_n^2 分布,因此可得拒绝区域为 $k>\chi_n^2(\alpha/2)$ 或 $k<\chi_n^2(1-\alpha/2)$,其中$\chi_n^2(\alpha)$ 为 χ_n^2 分布的临界值,可从χ^2 分布表查得。σ^2 的置信度为 $1-\alpha$ 的置信区间是

$$\left(\sum_{i=1}^{n}(X_i-\mu)^2/\chi_n^2(\alpha/2),\sum_{i=1}^{n}(X_i-\mu)^2/\chi_n^2(1-\alpha/2)\right)$$

(4)上述问题中若 μ 未知,则当 H_0 为真时,$ns^2/\sigma_0^2\sim\chi_{n-1}^2$,拒绝域为 $ns^2/\sigma_0^2>\chi_{n-1}^2(\alpha/2)$ 或 $ns^2/\sigma_0^2<\chi_{n-1}^2(1-\alpha/2)$。$\sigma^2$ 的置信度为 $1-\alpha$ 的置信区间为

$$(ns^2/\chi_{n-1}^2(\alpha/2),ns^2/\chi_{n-1}^2(1-\alpha/2)) \qquad (4)$$

比较两个正态母体方差的大小还有 F 检验法。

对于二元正态母体，可以用 t 检验法检验其相关系数是否为 0，也可用 χ^2 检验法检验两个分量是否独立。最常用的分布检验法是皮尔逊 χ^2 检验法、柯尔莫哥洛夫检验法、斯米尔诺夫检验法。

参考文献

1. 陈希孺. 数理统计引论. 北京：科学出版社，1981

2. 中山大学数力系. 概率论与数理统计（下册）. 北京：人民教育出版社，1980

（金治明）

jia tuoji

假脱机（simultaneous peripheral operations online spool） 使独占使用的设备变成可共享设备的虚拟设备技术。

假脱机技术是低速输入输出设备与主机交换的一种技术，通常也称为“假脱机真联机”，其核心思想是以联机的方式得到脱机的效果。低速设备经通道和设在主机内存的缓冲存储器与高速设备（通常是辅存）相联。为了存放从低速设备上输入的信息，或者存放将要输出到低速设备上的信息（来自内存），在辅存分别开辟一固定区域，叫“输出井”（对输出），或者“输入井”（对输入）。简单来说就是在内存中形成缓冲区，在高速设备形成输出井和输入井，传递的时候，从低速设备传入缓冲区，再传到高速设备的输入井，再从高速设备的输出井，传到缓冲区，再传到低速设备。

假脱机输入输出是利用快速、大容量、可为多个进程共享的磁盘设备作为中介，模拟多个非共享的字符输入输出设备。于是，需要独占使用这种字符设备的进程不再需要直接使用物理设备，而代之与这种虚拟设备进行信息交换，虚拟设备和物理设备之间则在系统特设的假脱机进程控制下进行信息传输。

应用假脱机技术的典型实例是将一台独享打印机改造为可供多个用户共享的打印机，具体过程是：系统对于用户的打印输出，并不把打印机分配给该用户进程，而是先在输出井中申请一个空闲盘块区，并将要打印的数据送入其中；然后为用户申请并填写请求打印表，将该表挂到请求打印队列上。当打印机空闲时，输出程序从请求打印队首取表，将要打印的数据从输出井传送到内存缓冲区，再进行打印，直到打印队列为空。

假脱机技术的特点是：①提高了 I/O 速度，从对低速 I/O 设备进行的 I/O 操作变为对输入井或输出井的操作，如同脱机操作一样，提高了 I/O 速度，缓和了**中央处理器**（CPU）与低速 I/O 设备速度不匹配的矛盾；②设备并没有分配给任何进程，在输入井或输出井中，分配给进程的是一存储区和建立一张 I/O 请求表；③实现了虚拟设备功能，多个进程同时使用一独享设备，而对每一进程而言，都认为自己独占这一设备，不过，该设备是逻辑上的设备。

参考文献

孙钟秀，等. 操作系统教程. 4 版. 北京：高等教育出版社，2008

（徐良贤）

jiancuo bianma

检错编码（error detection） 在数据传输过程发生错误（误码）的时候，能在接收端发现错误的编码。检错编码属于信道编码的一种，本质是增加通信传输的可靠性。

检错编码的过程通常需要在源码流中插入具有特殊意义的附加码元，从而达到在接收端进行检错的目的，此时源码流的实际传输速率降低，称为编码开销。将源码流的实际传输速率（或比特数）除以信道传输速率（总比特数）等于编码效率；不同的编码方式，其编码效率有所不同。

在检错编码方案中，“奇偶校验码”和“循环冗余编码”是两种应用较广的方案。

（1）**奇偶校验码**（parity check）是一种最简单的检错编码，它根据被传输的一组二进制数据中“1”的个数是奇数或偶数来进行校验。采用奇数的称为奇校验，反之，称为偶校验。采用何种校验是事先规定好的。例如，发送的原始 ASCII 字符 G（1110001），采用奇检验时，加入一个奇偶校验位“1”，即传输为“11100011”；接收方收到后，检查此 8 位二进制数中“1”的个数，如果是奇数则认为无误码发生；如果是偶数，则肯定有误码发生。采用奇偶检验时，若其中两位同时发生错误，则会发生检测不出错误的情况。因此对于高数据速率或者噪声持续时间较长的情况，由于可能发生多位出错，因此，奇偶检验有局限。

（2）**校验和**（checksum）是在数据处理和数据通信领域中广泛使用，且比较简单的一种检错编码，它根据被传输的一组数据分段（如将一段数据按照 8 位或 16 位二进制进行分段）反码求和产生附加检错码（校验和）来进行检错。例如，在互联网 IPv4 包传输中设置了包头的校验和检错字段，为了计算一个

IPv4 包头的校验和,发送方首先将校验和字段置为 0;然后,对包头中每 16 位进行二进制反码求和(整个包头看成是由一串 16 位的字组成),结果存在校验和字段中,进行发送。在接收方,同样对包头中每 16 位进行校验和计算,由于接收方在校验和计算过程中包含了包头中的检验和字段,因此,如果包头在传输过程中没有发生任何差错,那么接收方校验和计算的结果应该为全“1”。如果结果不是全“1”,那么接收方就丢弃收到的数据包。校验和被大量用于数据通信领域中保证数据的完整性和准确性。事实上奇偶校验也是校验和的一个特例。

(3) **循环冗余编码**(cyclical redundancy check, CRC)利用二进制除法及余数的原理,生成检错码完成误码检错的功能。其基本原理如下:采用多项式编码方法,收发双方事先约定的一个生成多项式 $g(x)$,将被处理的固定长度源数据帧 (frame) 视为一个 n 阶的二进制多项式 $t(x)$,由 $t(x)$ 除以 $g(x)$ 得到的余数称为 CRC 校验码,也称为帧检验序列 (FCS),将源数据帧加上帧检验序列形成发送帧发送出去。接收方将接收到的帧除以生成多项式 $g(x)$,若余式为零则认为传输无差错;若余式不为零则传输有差错。

CRC 校验可以 100% 地检测出长度小于等于 k(k 为 $g(x)$ 的阶数)的突发错误,所以 CRC 的生成多项式的阶数越高,误判的概率就越小。国际电报电话咨询委员会(CCITT)建议:2 048 Kbit/s 的 PCM 基群设备采用 CRC-4 方案,使用的 CRC 校验码生成多项式 $g(x)=x4+x+1$。根据应用环境与习惯的不同,CRC 又可分为以下几种标准:

CRC-12 = X12 + X11 + X3 + X2 + X + 1,

CRC-16 = X16 + X15 + X2 + 1,

CRC-CCITT = X16 + X12 + X5 + 1,

CRC-32 = X32 + X26 + X23 + X22 + X16 + X12 + X11 + X10 + X8 + X7 + X5 + X4 + X2 + X + 1

等。从性能和开销上考虑,CRC 检错均优于奇偶校验及校验和等方式。因此,在数据存储和数据通信领域被大量应用,著名的通信协议 X.25 的 FCS(帧检错序列)采用的是 CRC-CCITT,WinRAR、NERO、ARJ、LHA 等压缩工具软件采用的是 CRC-32,磁盘驱动器的读写采用了 CRC-16,通用的图像存储格式 GIF、TIFF 等也都用 CRC 作为检错手段。

参考文献

1. 胡道元. 计算机网络(高级). 北京:清华大学出版社, 1999

2. 王达. 网络工程师必读——网络工程基础. 北京:电子工业出版社, 2006

(周杰 张凌)

jiancai guocheng

剪裁过程(tailoring process) 对软件过程和活动实施剪裁的过程。将一选定模型以及相关标准应用于某一领域或具体软件项目,形成该领域的模型及标准或该软件项目的各个软件过程和活动。最初选定的模型是相对抽象、具有相对普遍性的,而所选标准是描述活动全集的,将它们针对某领域剪裁意味着形成该领域的相对特殊的模型及标准;将它们针对软件项目进行剪裁意味着形成该项目的软件过程和活动,这是剪裁过程的双重意义。

剪裁过程的基本内容包括剪裁过程的各项活动和剪裁的指导原则。剪裁过程的各项活动:

(1) 指明项目环境 指明影响剪裁的项目环境特征。如使用的模型和方法;系统和软件需求;机构的政策、步骤和策略;系统和软件的大小、重要性和类型;参加人员和合作伙伴的数量等。

(2) 收集信息 应向所有会影响剪裁的机构收集信息。用户、支持人员、负责签订合同的人员、潜在的投标者应参与剪裁。

(3) 选取过程、活动和任务 根据上述收集到的数据,决定要实施的过程、活动和任务。

(4) 将剪裁的决定及其理由编成文档 所有剪裁决定及这些决定的理由应编成文档。

剪裁的指导原则:剪裁可分为二级,第一级是根据应用领域的不同进行剪裁,第二级是根据每一个具体的项目或合同进行剪裁。这里给出实施第一级剪裁的一些主要原则。

(1) 开发过程的剪裁 ①对嵌入或集成在系统中的软件,应考虑该过程中的所有活动,并且必须明确开发者是否要实施或支持系统的活动;②对于独立的软件,关于系统的活动可能不需要,但应加以考虑。

(2) 和评价有关的活动的剪裁 评价可以分成以下 5 类。前面 4 类评价属项目级,最后 1 类属机构级。应根据项目或机构的范围、大小、复杂性和重要性相应地选取和剪裁这些评价:①过程内部的评价:由实行过程内指定任务的人员进行的评价;②验证和确认:由获取者、供应者或一个独立的合作伙伴根据项目从不同的深度验证和确认软件产品;③联合评审和审计:由评审方和被评审方联合评价软件产品和活动的状况;④质量保证:由和软件产

品开发没有直接关系的人员进行,目的是独立地保证软件产品符合合同要求并按制订的计划执行;⑤改进:由一个为了有效管理和自我改进其过程的机构进行,与项目或合同的要求无关。

(3) 剪裁和应用软件过程的若干考虑　以下是剪裁和应用软件过程时需指明、考虑的一些关键的项目特征:①机构政策:指明机构的政策,如使用的计算机语言,安全性和保密性,硬件储备要求,风险管理等。②获取策略:为项目指明获取策略,如合同类型,合同方和验证、确认代理的参与,获取者参与合同的程度,合同者能力的评价等。③支持的概念:指明支持的概念,如期望的支持程度、更改程度、获取者或供应者是否提供支持等。④生存周期模型:指明项目使用的生存周期模型,如瀑布模型、迭代瀑布模型、演化模型、螺旋模型等。⑤合作伙伴:指明参与项目的合作伙伴以及人员的数目。对一个有许多合作伙伴和数十或数百人参加的大型项目,需要加强管理和控制。内部和独立的评价、评审、审计等是大型项目的重要工具,而对小型项目,这些控制可以大大减少。⑥系统级特征:指明系统级特征,如子系统数目和配置项等。若系统有许多子系统或配置项,则开发过程应对每个子系统或配置项仔细进行剪裁,并应考虑所有的界面和集成方面的要求。⑦软件级特征:指明软件级特征,如软件配置项的数目、类型、大小以及软件的重要性、技术风险等。若软件有许多软件配置项、部件和单元,则开发过程应对每个软件配置项仔细进行剪裁,并应考虑所有的界面和集成方面的要求。⑧风险性:软件开发可以有技术风险。如果所使用的软件技术不成熟,所开发的软件太复杂或软件包含安全性、保密性或其他关键性要求,则需要有严格的规约、设计、测试和评价。这时,独立的验证和确认将显得更加重要。

参考文献

1. IEEE Standard for Developing Software Life Cycle Processes — IEEE Std. 1074—1991. 1991

2. ISO /IEC 12207:1995 Information Technology— Software — Part 1: Software Life-Cycle Process. 1995

(黄嘉启)

jiandan leixing

简单类型(primitive type)　其值不能进一步分解为更简单的值的类型。又称基本类型或初等类型。

程序设计语言中简单类型的选择与该语言预期的应用领域有关,一般包括实型、整型、字符型和布尔型,它们往往是语言预定义的类型。实型值是实现定义的有理实数的一个子集,整型值是实现定义的整数的一个子集,字符型值是实现定义的一个字符集,布尔型的值可由字面常量或预定义的常量标识符 false 和 true 来定义。有些语言还允许用户自行定义简单类型,如在 PASCAL 和 Ada 中,用户可自定义枚举类型和子域类型,枚举类型是由通过列举有限多个标识符来定义的,这里每一个标识符(枚举符号)即表示一个枚举值,子域类型是通过指定一个已有类型的子集来定义的。例如:

```
Type color = (red,green,blue);
     sub = 28...31
```

离散简单类型是其值与整型(或其一个子域)有一一对应关系的一种简单类型。在 PASCAL 和 Ada 中,离散简单类型的值可以用于诸如计数、分支选择和数组下标中,而在其他语言中,仅整型可用于此类操作。

参考文献

Watt D A. Programming language concepts and paradigms. Prentice Hall,1990

(金凌紫　陈涵生)

jiandan wangluo guanli xieyi

简单网络管理协议(simple network management protocol, SNMP)　设计用于管理 IP 网络设备的互联网标准协议,是 IETF(互联网工程任务组)定义的互联网协议簇的一部分。该协议提供了网络管理系统(NMS)对被管理设备的访问方法。利用 SNMP,网络管理系统可以远程管理支持该协议的设备,包括监视网络状态、修改设备配置、接收网络事件警告等。它由一系列的标准组成,可分为三个相关联的部分,即管理信息库(MIB)、管理信息结构(SMI)和 SNMP 协议本身。

管理信息库(management information base, MIB)定义了用数据变量表示的被管理对象的信息。例如,被管理对象自身的描述信息,如系统名称、系统位置、联系方式等以及端口统计信息,包括接口速率、MTU、发送的报文数、收到的报文数等,可以用标量数据表示,即每个信息可用一个字符串或数表示。由 RFC 1213 定义的最基本的 MIB II 标准规定了一些常用的变量,要求所有支持 SNMP 的被管理端都要实现。其他类型的信息如桥协议、无线网络、BGP 路由协议等由一些特定的管理信息库分别定义。厂商和个人还可以定义私有 MIB 以扩展能够提供的

信息。被管理设备由内置的 SNMP 实体维护相关的变量数据，网络管理系统通过查询或设置这些变量实现对被管理对象的访问。

管理信息结构（structure of management information, SMI）定义了 SNMP 的基本数据类型和对管理信息库内容进行访问的规则。基本数据类型规定了变量表示的方式和取值范围，如接口 MTU 值是一个整数，用 INTEGER 类型表示；设备 CPU 利用率百分比最大不会超过 100，用 Gauge 类型表示。这里的 INTEGER 和 Gauge 就是 SMI 定义的基本数据类型。SMI 还设计了 SEQUENCE 和 SEQUENCE OF 两种基本数据类型，用以定义复杂的数据类型，定义方式类似于 C 语言中的结构。

管理信息结构定义了另一种特殊的基本数据类型：对象标识符（object identifier, OID）用于对被管理对象进行唯一标识。它的组织方式是树状结构，类似于 DNS。MIB 中定义的内容通过 OID 进行索引。具体地说，OID 由树上的一系列整数组成，整数之间用点（.）分开，每个节点也对应了文本名字以方便人们阅读。例如，1.3.6.1.2.1.1.5（或 iso. org. dod. internet. mgmt. mib-2. system. sysName）表示了访问被管理对象的系统名称的方法。在不引起歧义的情况下，大多数 SNMP 的应用程序支持用文本名字的一部分来进行访问，如用 sysName 来访问前述的系统名称对象。

SNMP 协议本身定义了网络管理系统与被管理对象之间进行信息交互的规范。版本的发展经历了 SNMPv1、SNMPv2 和 SNMPv3，其中 SNMPv2 的分支 SNMPv2c 是稳定成熟的版本，得到了较为广泛的支持。SNMPv3 主要解决了早期版本在安全性方面的问题，在安全性和可管理性方面进行了改进和增强，并对原有的术语和概念进行了修订和规范。SNMPv3 在架构上由多个不同功能的模块组成，支持 SNMPv1 和 SNMPv2 的所有操作，可以与旧版本并存。正因为如此，SNMPv3 正受到越来越多的支持和使用。

按照 RFC 2571 的规范，被管理端和管理系统端都被统称为 SNMP 实体，每个实体由一个 SNMP 引擎以及一个或多个 SNMP 应用组成。SNMP 引擎实现如下功能：发送和接收报文、准备和提取报文内容、认证和加密消息、对被管理对象进行访问控制等。这些功能以服务的形式提供给 SNMP 应用，SNMP 应用完成网络管理的各种任务，如命令发生、命令响应、产生通知、接收通知、消息中继等。

SNMP 的操作可分为两类，一种是进行查询或修改，由网络管理系统发出请求，被管理对象给出响应；另一类是主动通知，通常由被管理对象发出。在 SNMPv2 和 SNMPv3 中还支持消息的中转，即操作可以由一个管理系统发往另一个管理系统。操作的内容采用规定的标准协议数据单元（protocol data unit, PDU）进行传送。SNMP 定义的 PDU 中，getRequest、getNextRequest、getBulkRequest（SNMP v2 和 v3 支持）、setRequest 属于命令请求类，由网络管理系统发往被管理端，用于查询或修改被管理对象的值。Response 响应查询和设置操作，通常运行在被管理端，属于命令响应类。被管理端产生的事件消息，如设备加电重启、设备的端口状态改变、设备的硬件配置发生改变等，可由被管理端主动进行发送，Trap 和 InformRequest（SNMP v2 和 v3 支持）都能发送这类的信息，所不同的是 Trap 是单向的，不要求回应，而 InformRequest 发送后管理端需要给出 Response 进行回应。

作为一种应用层协议，SNMP 通常封装在 UDP 报文中，因此不能保证报文在传输过程不丢失或发生错误。另一方面，UDP 不要求保持连接，对系统的开销较小。为了能在网络上进行传输，SNMP 使用 BER（basic encoding rules）作为编码方案，所有定义的对象数据都被编码成八位字符，从而保证在网络上进行传输后能被正确解码。SNMP 默认采用 161/udp 和 162/udp 端口分别作为 SNMP 请求和 Trap 的端口。SNMP 对网络层无要求，因此也能支持 IPv6 作为网络层协议，但要求两端都同时支持 IPv6 进行传输。

简单网络管理协议具有结构简单、功能强大、易于扩展等优点，在网络管理领域得到了广泛的接受，已经成为事实的国际标准。另外，RMON 远程网络监控协议在 SNMP 的基础上进行了扩展，成为了网络性能和故障监测的重要手段。大量的网络设备和系统都提供了 SNMP 协议的支持，一些开源的网络管理工具软件如 MRTG、Cacti 等，使用 SNMP 协议对众多网络单元进行管理，功能强大，易于使用。

参考文献

1. RFC 1213—Management Information Base Ⅱ

2. RFC 2571—Architecture for SNMP Frameworks

3. SNMP 工作组：http://www.ietf.org/wg/concluded/snmpv3

4. SimpleWeb: http://www.simpleweb.org/

(周昌令)

jianma

键码(key) 实体的一个属性或一组属性,其值可用来唯一标识该实体。键码又称作键或码。

在数据库领域中,当用概念模型描述现实世界客观事物时,将客观存在并可互相区分的事物称作实体,实体所具有的特性称作属性,能够唯一标识实体的一个或一组属性称作键码。例如学生实体可有学号、姓名、性别、所在系、年级等属性,其中学号是键码。

在关系数据库系统中,基于**数据依赖**概念可以更严格地定义键码如下:设有关系模式 R$\langle U, F\rangle$,其中 U 为属性集合,F 为 U 上的函数依赖集合。被 F 逻辑蕴涵的函数依赖的全体构成的集合,称为 F 的闭包(closure),记为 F^+。若有属性组 $K \subseteq U$,满足 $K \to U \in F^+$,而 K 的任何真子集 K' 都不能满足 $K' \to U \in F^+$,则称 K 为 R 的候选键码(candidate key)。当候选键码多于一个时,则选定其中一个为主键码(primary key)。属于任一候选键码的属性称为主属性。

键码是**数据库系统**中的重要概念。它是信息检索的重要依据。给定候选键码值作为检索条件,则检索的结果是至多一个记录或一条元组值。

在关系数据库系统中,如果关系 R_2 的一个属性(或属性组)的值与另一个关系 R_1 的主键码的值相对应,则称 R_2 中的这个属性(或属性组)为外键码。(说明,R_1 和 R_2 不一定是两个不同的关系)。例如,对于下面两个关系

部门(部门号,部门名,部门地址)

职工(职工号,职工名,部门号,主管领导职工号)

“职工”关系的“部门号”属性与“部门”关系的主键码“部门号”相对应,因此“职工”关系中的“部门号”为外键码。在“职工”关系中,“主管领导职工号”属性与主键码“职工号”属性相对应,因此“主管领导职工号”是“职工”关系的外键码。

关系数据库中外键码与主键码的对应是关系之间联系的体现,它反映了一个关系中的元组对另一个关系中的元组的参照完整性约束关系。

代理键码(surrogate key)是替代键码和候选键码的人工键码,它一般采用自然数序列,不是实体所具有的实际属性,是具有唯一性的人工键码。当关系的键码为一组属性或长文本类型属性时,代理键码能够简化键码数据类型,从而减少检索和连接操作的键码匹配代价。代理键主要应用于数据仓库中,作为维度主键的一个替代键,用于标识数据仓库中较少更新应用模式下缓慢变化的维度。

参考文献

1. Date C J. An introduction to database system. 8th ed. Reading: Addison-Wesley Publishing Company, 2005

2. Ullman J D. Principles of database and knowledge-base systems. Vol. 1. Computer Science Press, 1990

3. 王珊,萨师煊. 数据库系统概论. 4 版. 北京:高等教育出版社,2006

(唐世渭 杨冬青 王珊)

jianpan

键盘(keyboard) 按有序排列组成的并带有功能电路的一组键体开关。使用者通过击键向计算机输入程序、命令、数据等,是人对计算机进行控制的重要工具。

计算机标准键盘的键数,少的有 83 个,多的可达 105 个,视使用要求而异。键盘可划分成三个区。中央为打字键区,包括字母、数字、符号和一些特殊功能键,如换档键、回车键等。键的排列和标准打字机相同,这种排列不一定合理,但已习惯使用,约定俗成。从 1868 年以来就没有变化。已有 ISO 2530 和国家标准 GB 2787 规定。右侧为数字键区,用于数字输入和进行四则运算,也可作屏幕编辑、移动光标使用。第三个区设置在键盘的左边或上方,是 10～12 个特殊功能键,其具体功能由使用者定义。一些多媒体计算机的键盘还增加一些快捷键。

图 1 是计算机增强型键盘的键盘排列图。键盘的每个键有一个键开关,键开关有机械触点式、电容式、薄膜式等多种。其作用是检测出使用者的按键动作,把机械的位移转换成电信号,输入到计算机中去。

键开关电路排成一个 $m \times n$ 的矩阵。某一字符的键开关在 $m \times n$ 矩阵上的位置是固定的。例如 IBM-PC 机的键盘,字符 a 的键位是 $m = 14, n = 2$。

键盘内部有键盘控制器,它由**单片计算机**和一些芯片组成。单片计算机进行扫描,判断按下键的位置,保存键的键位码,并输入到计算机中去。

标准键盘功能强,除了可输入英文和拉丁文字外,也可用作汉字输入,不同的是键位码代表的不是英文字母而是汉字的拼音字母、字形或音节。它由

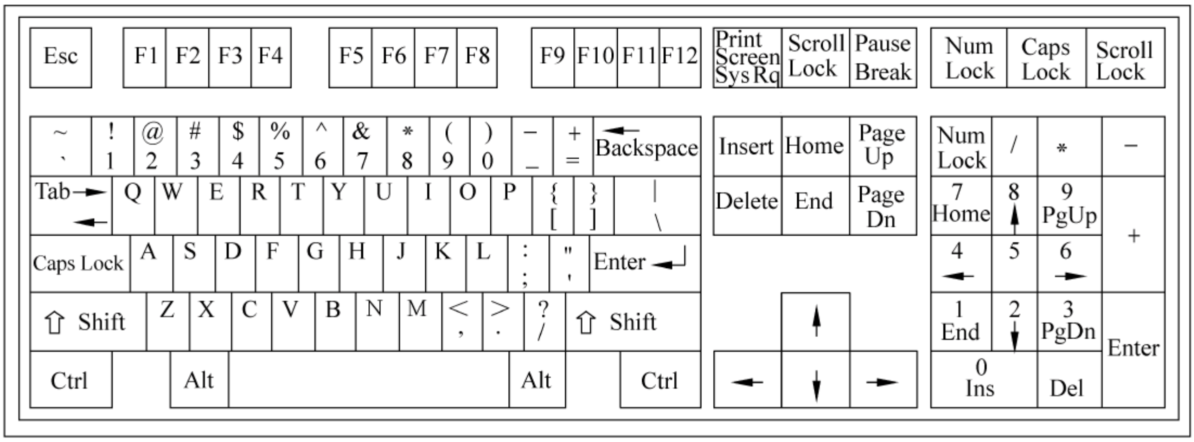


图1　键盘排列图

各种汉字输入方法自行定义。键位代码送入计算机后经过软件处理，转换成机内码，再进行其他处理。

计算机和终端设备所用的键盘是一个独立的部件，通过电缆和主机连接。20世纪90年代以来，无线的红外线传输技术和蓝牙传输技术也有应用。键盘的标准外形是长方形。分成左右两半的人体工程学键盘也广泛应用。

由键盘输入的内容通常由显示器屏幕即时显示出来，因此非常直观。键盘价廉、轻便。但随着**鼠标**、控制球、**手写输入板**、**触屏**等输入设备的不断普及，在某些方面取代了键盘。但键盘仍是人机交互式输入设备中的一个重要设备。　（林兼）

jiangshi wangluo

僵尸网络（botnet）　攻击者（botmaster）出于恶意目的，传播僵尸程序（bot）控制大量联网主机所组成的覆盖网络。僵尸网络的特征是攻击者能够通过一对多的命令与控制信道操纵感染僵尸程序的主机执行相同恶意行为，如可同时对某目标网站进行分布式拒绝服务攻击。

僵尸网络活动主要分为传播、命令与控制和攻击三个阶段。

僵尸网络可采用各种恶意代码传播方式向联网主机植入僵尸程序，包括主动扫描并远程漏洞攻击、弱口令扫描入侵、邮件病毒、网页挂马攻击、社会工程学等。

僵尸网络的命令与控制机制决定了其拓扑结构、通信效率、可扩展性和自身安全性，是僵尸网络工作机制的核心。按照命令与控制机制使用的通信协议，僵尸网络可进一步分为IRC僵尸网络、HTTP僵尸网络、P2P僵尸网络和自定义协议僵尸网络等类型。

发动网络攻击是攻击者部署僵尸网络的最终目标，具体攻击方式包括分布式拒绝服务、发送垃圾邮件、网络钓鱼、单击欺诈以及敏感信息窃取等。

僵尸网络采用了加密通信、身份认证、动态邻居发现、动态域名（domain-flux）、IP地址快速迁移（fast-flux）、内核套件（Rootkit）、反调试等技术增强对抗性。

僵尸网络历史渊源可追溯到1993年在IRC网络中出现的良性Bot——Eggdrop，1999年PrettyPark与SubSeven恶意代码开始使用IRC协议搭建一对多的命令与控制信道，成为最早的僵尸网络。此后出现了GT-Bot、SDbot、Agobot等知名IRC僵尸程序，IRC僵尸网络成为早期最主要的僵尸网络类型。为了达到更强的隐蔽性和对抗性，攻击者不断对僵尸网络的组织形式进行创新和发展，出现了以Bobax为代表的HTTP僵尸网络，以Sinit、Phatbot、Storm、Nugache为代表的P2P僵尸网络及以Mariposa为代表的自定义协议僵尸网络。

由于可为攻击者提供隐匿、灵活和高效的攻击控制平台，僵尸网络已快速发展成为目前互联网上最严重的安全威胁之一，近年来频繁出现了Storm、Conficker、Zeus等超过数百万节点的超大规模僵尸网络。此外僵尸网络也是分布式拒绝服务、垃圾邮件等安全威胁的重要来源。

目前互联网终端主机主要依赖传统反病毒技术来检测和防御僵尸程序，避免被僵尸网络控制。而从根本上遏制僵尸网络需要安全应急响应组织、法律机构、厂商、ISP多方协作，采用域名“黑洞”、关停

命令控制服务器等技术手段，遏制僵尸网络发展甚至直接摧毁整个僵尸网络，已有成功案例包括针对Waledac、Rustock 等大规模僵尸网络的应急处置。

针对僵尸网络的准确检测、监测分析和防御方法是信息安全学术界的研究热点，并已取得了丰富研究成果，但彻底消除僵尸网络威胁仍具有很大的挑战。

参考文献

1. 诸葛建伟，韩心慧，等. 僵尸网络研究. 软件学报，2008，19(3)

2. 江健，诸葛建伟，等. 僵尸网络机理与防御技术. 软件学报，2012，23(1).

（诸葛建伟）

jiaocha bianyi chengxu

交叉编译程序（cross compiler） 本身在机器 A 上运行但生成另一台机器 B 上的目标代码的编译程序。这两台机器中前一台 A 称为交叉编译程序的**宿主机**，后一台 B 称为交叉编译程序的**目标机**。

假定语言 L 是一种适合写编译程序的语言，且在机器 A 上有 L 的一个交叉编译程序，它生成机器 B 上的目标代码，该交叉编译程序的 T 图如图 1 所示，或写为 L_AB（参见**自编译程序**），可以利用它来实现机器 B 上的任一语言，如 FORTRAN 的编译程序。首先用 L 号机器 B 上的一个 FORTRAN 的编译程序 $FORTRAN_LB$，然后用 L_AB 对 $FORTRAN_LB$ 进行编译，便可得到机器 B 上的 FORTRAN 编译程序 $FORTRAN_BB$ 如图 2 所示。

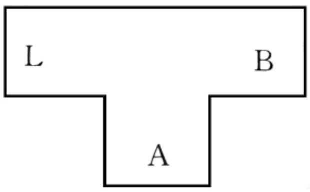


图 1 交叉编译程序的 T 图

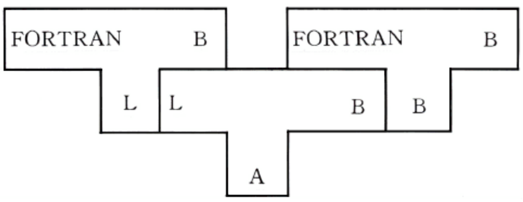


图 2 交叉编译过程一例

交叉编译技术的主要用处是可以在较大型的具备必要的软件开发工具的宿主机上为小型的或软硬件配置简单的目标机开发高质量软件给予支持。交叉编译技术还可以在现有的机器上为正在制造中的新机器上的软件开发给予支持。

参考文献

Aho A V，Sethi R，Ullman J D. Compilers principles，techniques，and tools. Addison-Wesley，1986

（徐永森）

jiaohushi dianshi

交互式电视（interactive television） 观众通过遥控器和菜单可以选择和控制节目的电视。它是一种非对称双工通信模式（参见**异步传输**）的新型电视业务。交互式电视分两种：节目间交互式电视和节目内交互式电视。

节目间交互式电视又称为点播电视（VOD），它分为真点播电视（TVOD）和准点播电视（NVOD）。真点播电视要求电视台（视频服务器）对每个用户的点播请求都要及时响应，允许用户对装设在信息中心和电视台视频盘和视频带上的节目可以随意控制。准点播电视是每隔一定时间（如 10 分钟）从头播放一套节目，用户点播某个电视节目时，交换机将终端与最近将要从头开播的频道连通，用户等待时间不会超过时间间隔。等待时间可以播放广告和片花，增加收入，降低点播费用，现在应用较多。

节目内交互式电视也称全交互型电视，它能够将电视台（视频服务器）对用户的请求应答即时传送给用户，传送给用户的信息内容包括视频、图像、语音、文字和数据。

交互电视可以应用在点播电视、教育、远程购物、交互式游戏、新闻点播服务等领域。交互电视的应用一般采用非对称通信模式，即下行宽带传输，上行窄带传输。应该指出的是，在这些应用中，导航和内容的无缝集成很重要。

交互电视系统核心功能应包括传输、会话、访问控制、导航与节目选择、应用启动、媒体同步链、应用控制、表现控制和用户概况等。交互电视系统主要的组成部分有视频服务器、编码器-路由器、用户请求计算机和记账计算机以及位于用户端的机顶盒。其中，视频服务器是交互式电视系统中最关键的组成部分，它的性能直接影响服务的质量。它可以是基于个人计算机（PC）和工作站的机群结构，也可以采用大规模并行计算机结构。

作为交互电视系统的中央控制和服务部分，视频服务器负责解决大容量视频存储、节目检索和服

务、高吞吐量视频传输等需求,它具有以下功能:

（1）请求处理:接收用户的访问请求;

（2）许可控制:检查用户的权限,考虑新请求的加入是否影响已有服务的性能;

（3）节目检索:从服务器的存储系统中检索节目的存放位置;

（4）可靠的流传输:向用户提供一个实时的数据流;

（5）支持录像机功能:如暂停、启动、快进、快退等功能。

机顶盒(STB)是一种与电视机接口并提供附加服务的设备。交互式电视的机顶盒应具有以下功能:①提供与网络的接口;②音频与视频的解码;③用户界面和图形控制;④外围设备控制;⑤安全与权限管理。

交互式电视系统的特点和设计中应该考虑的问题包括:

（1）信息通信的不对称模式,这是交互式电视系统与分布式多媒体数据库系统的一个重要区别。交互式电视系统和人机间直接通信系统一样,数据的传送量和接收量之间有大量的差异。人类的眼脑和耳脑结合可以迅速地接收非常多的信息,而操作键盘和定位设备的手就要慢几个数量级。使用单键遥控器的交互式电视系统,目前要以每秒兆位的速度传送家用质量的电视,而从遥控器到机顶盒只是每分钟几位的数量级。

（2）可扩展性和模块化。无论是有线电视和电话公司的 VOD 系统,还是蜂窝电视和宾馆点播电视 VOD 系统,都有一个用户扩充和技术更新的问题。要有效地解决上述问题,系统必须设计成模块化,使其具有可扩展性。

（3）选择节目内容和选择时间的集中性。根据目前示范系统的经验,在一个星期中,交互式电视系统大多数节目的选择集中在一个非常小的节目子集中。另一个值得重视的特点是节目选择时间的集中性。

交互式电视最常用的是节目间的交互,亦称点播电视(VOD)。典型的 VOD 系统主要由下面四部分组成:视频服务器、编码器/路由器、用户请求计算机和记账计算机及机顶盒。其中有两个关键技术:①视频服务器。VOD 系统为了进行点播电视服务,需要一个大容量存储器和具有高速传输能力的快速检索系统,这就是视频服务器。②电视机机顶盒。电视机机顶盒是 VOD 系统用于节目选择与 VOD 系统交互通信的用户室内设备。

参考文献

钟玉琢,蔡莲红,史元春,沈洪. 多媒体计算机技术基础及应用. 3 版. 北京:高等教育出版社,2009

（钟玉琢,孙立峰）

jiaohushi yuyan

交互式语言(interactive language)　支持使用者与计算机系统通过交互应答完成计算任务的语言。又称**会话语言**。

交互式语言最大的优点在于语法简单、易读性强、动态性高,许多交互式语言还具有很强的支持跨平台的特性。交互式语言和非交互式语言最大的区别在于是否支持在程序运行过程中进行编程。非交互式语言程序需要先写成文档,然后编译,最后运行编译后的程序。而交互式语言程序执行过程中可以请求用户输入数据,甚至修改程序、指出下一步所要采取的动作。用户也可以中止程序的执行,待完成其他工作后再从程序中断点接着往下执行。交互式语言程序执行停止时,只要系统仍处在该语言的执行环境中,则运行程序所设置的状态(变量及其所赋的值等)保持不变。

交互式语言程序一般通过解释方式执行。用户及程序人员可以在同一环境下输入、调试、运行程序,这使程序开发比较容易,程序设计环境对用户也比较友善,但也因此影响程序执行功效和内存使用功效。

许多数据库语言(如 SQL)实际上就是交互式语言,**BASIC 语言**是典型的交互式语言,Smalltalk、Python 等动态语言也是交互式语言。

参考文献

1. INCITS/ISO/IEC 10279-1991(R2005), Information Technology-Programming Languages-Full BASIC, 2005

2. Goldberg A, Robson D, Smalltalk 80: The Language. Addison-Wesley, 11 January 1989

3. Python Software Foundation. Python 3.0 Release. http://python.org/download/releases/3.0/, 2009

（徐宝文）

jiaohuanji

交换机(switch)　由输入输出接口以及具有交换分组或信元等数据单元能力的转发逻辑组成的网络

设备。转发逻辑描述了用于交换技术的规则，实现数据单元的转发。输入输出端口是物理的和逻辑的接口，用来连接到需要交换数据单元的通信网络。

交换机在网络中具有非常重要的作用，它能够基于 MAC 地址识别完成数据包的转发功能。由于交换机只会把数据包传送到目的节点，而不是广播到所有节点，因此相对集线器而言，交换机不会浪费网络资源，同时保证数据安全传输。

交换机的种类繁多，根据网络覆盖范围划分，有广域网交换机和局域网交换机；根据传输介质和传输速度划分，有快速以太网交换机、千兆以太网交换机和万兆以太网交换机等；根据应用层次划分，有核心层交换机、汇聚层交换机和接入层交换机；根据结构划分，有固定端口交换机和模块化交换机；根据工作的协议层划分，有第二层交换机和第三层交换机，其中第二层交换机更为常见，转发逻辑如图 1 所示。

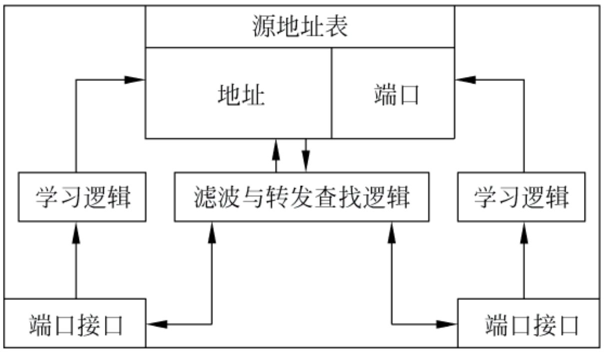


图 1　第二层交换机的转发逻辑

根据图 1，第二层交换机可以记住每一个接口收到的数据帧的源 MAC 地址，并且通过学习逻辑将该地址和对应的端口存入内存中的源地址表中。而当某个接口上收到数据帧时，交换机会通过滤波与转发查找逻辑在源地址表中查找到目的 MAC 地址所对应的输出端口。

交换机有两种交换技术，一种是存储转发，另一种是穿通传送。

存储转发技术要求交换机在转发数据帧至目的段以前，接收并缓存整个数据帧。这样可使交换机在传递数据帧至目的段以前校验识别并丢掉有差错的数据帧。缓存所带来的延迟是这种技术的主要缺点。

穿通传送技术的目的就是减少存储转发中的延迟。这种技术不需要缓存整个数据帧，而只需缓存用以决定目的地址的帧头，然后就将数据帧很快地直接送到目的地。其主要缺点是无法有效地检查出坏的数据帧。

由于这两种技术各有利弊，现在也出现了结合两种技术的交换机。如可以根据网络运行情况自动选择不同交换技术的混合型交换机，以及采用折中方式，缓存数据帧的前 64 位字节校验后再传送的改进型穿通传送交换机。

随着交换技术的发展，第 3 层交换机也逐步成熟起来。该类交换机将传统路由器的分组处理功能和交换机的速度优势结合在一起，可以同时在 OSI 七层网络模型的第二层和第三层上工作，也被称为多层交换机或路由交换机。当网络正常运行时，为了获取更快的速度，多层交换机在第二层工作。当网络流量超出正常范围时，多层交换机改用路由功能处理网络流量。这样，第三层交换机避免了不必要的网络路由成本，可以根据不同的网络状况实现最优化的网络性能。

然而，三层交换机主要用于局域网或城域接入网的数据交换，路由器主要用于广域网或园区出口的 IP 数据包的路由和转发。因此针对主要需求的不同，设备的结构和设计也不同。三层交换机虽然具备了一些基本的路由器功能，但仍是以太网交换机。在实现上，三层交换机在对第一个数据流进行路由后，它将会产生一个以太网 MAC 地址与 IP 地址的映射表；当同样的数据流再次通过时，将根据此表直接从以太网二层转发而不是再次查找 IP 三层转发表。而路由器完全是按照 IP 三层转发表的查找来设计的。三层交换机专注转发性能，其端口一般均为以太网端口，交换矩阵采用包交换，吞吐量大，但三层转发表项条目少、附加功能少、抗网络震荡能力弱。而路由器端口类型丰富，交换矩阵采用信元交换，三层表项条目多，附加功能多，抗网络震荡能力强。

参考文献

1. 胡道元. 计算机局域网. 3 版. 北京：清华大学出版社，2002

2. 苗凤君. 局域网技术与组网工程. 北京：清华大学出版社. 2010

（毕军）

jiaohuan jishu

交换技术（switching technologies）　采用交换机或接点机设备和有限线对，通过路由选择，在要求通信的双方之间建立物理的或逻辑的连接，形成通信电路，以实现通信双方话音或数据传输的技术。

交换技术起源于电话交换系统，一直到 20 世纪 70 年代，当人们说到交换，指的仍然是电话交换。计算机网络的兴起和发展，就有了数字交换的概念和一系列飞速发展的交换技术。现在通常采用的交换技术有**电路交换**和**分组交换**等。

参考文献

1. 张元，等. 从局域网到广域网——九十年代的网络管理技术. 北京：北京市新闻出版局，北京希望电脑公司，1991

2. Stalling W. Local and metropolitan area network. 4th ed. New York：MacMillan Publishing Company，1993
（史美林　徐明伟）

jiekou

接口（interface）　系统的（硬件或软件）组件和其他组件的交互点。接口使得其他组件可以在不了解该组件内部结构的情况下和它进行交互，完成特定的功能。为了能够保证同一类型的组件之间可以替换，人们会定义一些标准接口。遵守标准接口的组件可以互相替换，给生产者和使用者都带来方便。

硬件接口是指计算机的各个硬件组件（比如总线，内存，显示器，输入输出设备）之间的交互点，例如 SCSI 接口标准。根据逻辑层次的不同，这些接口可以通过底层电气信号特性和/或抽象逻辑信号进行描述。外界和组件的交互过程必须遵守特定的协议，这些协议定义了组件和外界进行信号交换时必须遵守的顺序和规则。

软件组件的接口指定了它和其他软件组件之间的交互方式。软件接口的重要作用是把组件的具体实现隐蔽起来（称为封装）。组件的使用者只需要了解接口，而不需要知道组件的具体实现。只要两个组件具有相同的接口，即使它们的实现不同，也可以互相替换。

根据软件组件的不同粒度，软件接口有不同的层次。

对于程序中的一个函数或者过程，其接口就是这个函数或过程的范型，它通过参数类型和回送类型来描述调用这个函数或过程的方法。在 Eiffel 语言中，人们还可以通过前/后置条件在语义层次上描述函数的接口，告诉程序员调用这些函数必须满足的语义条件以及这些函数能够完成什么任务。

对于一个包含多个函数的模块，它的接口通常包括这个模块中定义的常量、数据类型、模块中各个函数的首部以及异常定义等，有时也会把模块中定义的一些公共变量当作接口，但是这样做会破坏模块的封装特性。通常人们还会描述模块中各个函数之间的相互关系以及使用这些函数的模式，但是这些描述多数是非形式化的，因此编译程序不能进行自动检查。

在面向对象的程序设计中，类是组成程序的基本组件。一个类的接口包含一组成员变量和成员函数（或称例程），它指定了外界使用这个类的方式。为了进一步提高封装性，类的接口还指定了各个接口元素的可见性，不同的使用者（其他类、子类）可见的接口有所不同。

在 Java 语言中，接口是一个独立的概念，它定义了一组**例程**的范型，但是没有给出实现。一个接口可以由多个**类**实现，同时一个类也可以实现多个接口。类实现接口的条件是它实现了接口中规定的所有例程。Java 中的接口实际上是一个数据类型。如果一个类实现了某个接口，那么这个类的对象就可以被赋值给类型为该接口的变量。

对于更大粒度的软件组件，比如**操作系统**、**数据库**、中间件平台等，它们的接口更加复杂。这样的接口会包含很多复杂内容，比如应用程序接口、特定的描述语言、特定的编程模式等，数据库系统的接口包含了结构化查询语言 sql 的标准。

参考文献

1. ISO/IEC 9316. Information technology—Small Computer System Interface-2. 1995

2. ISO/IEC 25436. Information technology—Eiffel：Analysis，Design and Programming Language. 2006

3. Schildt H. Java The Complete Reference，8th ed. McGraw-Hill Osborne Media 2011
（赵建华）

jiekou dingyi yuyan

接口定义语言（interface definition language）　描述对象与其外部世界接口的语言。接口是服务的提供方和使用方之间互相合作的约定，它仅描述如何使用的信息，不包括具体实现信息，接口体现了“信息隐蔽”的原则。

接口定义语言与编程人员通常使用的**程序设计语言**都是计算机语言。接口定义语言独立于具体的程序设计语言，但可以向它自动映射，例如：生成相应的类型说明，生成相应的代理机制等。这对于快速开发客户程序十分有利。

接口定义的核心是对操作的描述。通常将对操作的一个形式化描述称为操作的标记。随着网络软

件的发展,对操作描述的内容也在逐步完善。最简单的操作仅包括功能性信息,复杂的操作还包括约束性信息。

(1) 功能性信息 接口的功能性信息是对接口中各个操作的功能描述。操作是由操作符标识的实体,它指明了一个不可再分的服务原语。与数学上一个函数的映射过程类似,对一个操作功能的描述主要由输入、输出两部分组成,分别用于描述操作的输入、输出参数名称及类型。例如:操作标记 int add(in int a 1, in int a 2)定义了一个加法操作,该操作的输入是两个整数,输出是一个整数。

(2) 约束性信息 接口的约束性信息是指对功能约束的描述。网络环境下不同软件模块之间的合作需要考虑的因素不仅包含功能方面,还涉及分布性、可靠性、安全性等方面的因素。只有了解了这些因素,不同的软件才能真正合作起来。因此,网络环境下的软件接口除了需要定义模块的功能性内容外,还需要定义模块的约束性内容。简单的约束包括:异常处理、时间限制、客户身份、可靠性要求等。复杂的约束包括:①行为特征,用于描述操作的外部特征,这一般是通过对操作增加前置与后置条件而实现的;②同步特征,用于描述操作的并发性等特征。

OMG 组织提出的接口定义语言(OMG-IDL)是较早出现的接口定义语言。OMG-IDL 于 1991 年提出,并且已被 ISO 组织采纳(ISO DIS 14750)。后来陆续发展的接口定义语言还有:微软的 MIDL,Web 服务的 WSDL 等。

由于接口较好地分离了相互协作的软件,使它们可以分别演化,比较好适应变化,因此受到了软件研究人员的广泛关注。

参考文献

OMG. The Common Object Request Broker: architecture and specification, 2.3.1.1999 (王千祥)

jiegouhua buxian xitong

结构化布线系统(structured cabling system, SCS) 一种模块化、灵活性极高的建筑物和建筑群的信息传输系统。结构化布线系统使用一套由共用配件所组成的配线系统,将所有语音、数据、图像及多媒体业务的设备的布线网络组合在一套标准的布线系统中,实现公共电信配套设施的一次性建设。在国家标准《建筑与建筑群综合布线系统工程设计规范》(GB 50311—2007)中,称为**综合布线系统**。

结构化布线系统的发展与建筑物自动化系统密切相关。传统布线系统如电话、计算机局域网都是各自独立的,各系统分别设计和安装,采用各种不同的传输线缆、配线插座和连接器件,例如传统电话采用 1 对或 2 对的双绞线,而计算机网络则采用同轴电缆、4 对无屏蔽双绞线和光缆。当用户需求发生变化时,各系统均需要增加或调整相应的端口,并改变相应线缆的铺设。在结构化布线系统中,当终端设备的位置需要变动时,只需做一些简单的跳线,而不需要再布放新的电缆以及安装新的插座。

结构化布线系统应为开放式网络拓扑结构,应能支持语音、数据、图像、多媒体业务等信息的传递。结构化布线系统采用开放式星型拓扑结构,每个子系统都是相对独立的单元,对每个子系统改动都不影响其他子系统。

结构化布线系统由以下 6 个子系统组成(见图 1):工作区子系统;水平子系统,又称配线子系统;干线子系统,又称垂直子系统;建筑群子系统;设备间子系统;管理子系统。

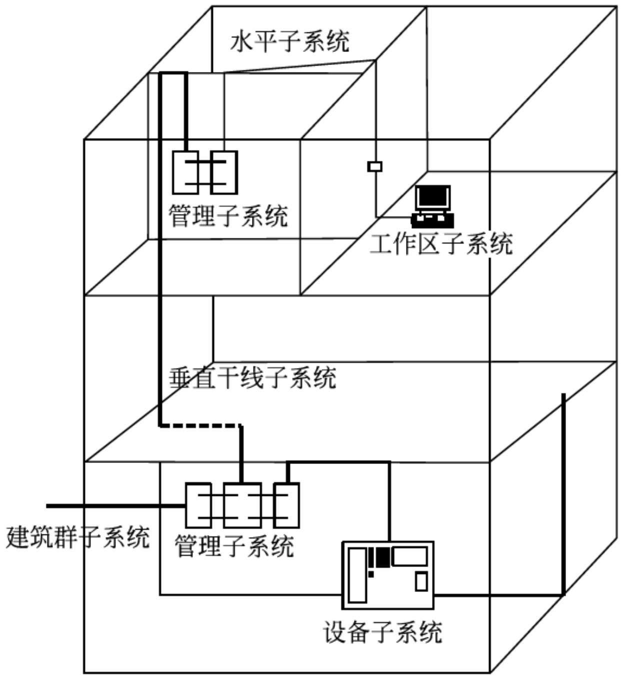


图 1 结构化布线系统的系统结构图

传统结构化布线标准在进线部分没有明确定义,仅仅是在建筑群子系统做了简单的说明。随着社会发展,拥有大规模的建筑群越来越多了,建筑群之间的进线设施已经越来越不可忽视。国家标准《建筑与建筑群综合布线系统工程设计规范》(GB 50311—2007)将进线间独立为一个子系统,规

定了进线间是建筑物外部通信和信息管线的入口部位,作为入口设施和建筑群配线设备的安装场地,同时明确了相应的功能。

参考文献

1. ANSI/TIA-568-C. Generic Telecommunications Cabling for Customer Premises

2. GB 50311—2007. 建筑与建筑群综合布线系统工程设计规范. 2007

(尚群)

jiegouhua chengxu sheji

结构化程序设计(structured programming) 具有结构性的编程方法。结构性主要反映如下:第一,编程工作为一演化过程,即按抽象级别依次降低、逐步精化,最终得出所需程序的方法编程(自顶向下逐步精化);第二,按模块组装的方法编程,亦即,将所需程序编制成由若干模块(或构件)组成;第三,将所需程序编制成只含顺序构造、判定构造以及重复构造,其中每种构造只允许单入口和单出口。

采用结构化程序设计方法编程,旨在提高编程质量与所编程序的质量。自顶向下、逐步精化方法有利于在每一抽象级别上尽可能保证编程工作与所编程序的正确性;按模块组装方法编程以及所编程序只含顺序、判定、重复三种构造则可使程序结构良好、易读、易理解、易维护,并易于保证及验证程序的正确性。

1964 年 C. Bohm 与 G. Jacopini 曾证明,任一流程图均可利用重复与嵌套等价地改写成只含可执行语句列、判定子句和迭代构造,并且每种构造只有一个入口和一个出口。1968 年 E. W. Dijkstra 在给 ACM 通讯的一信(该信由 ACM 编辑添上如下标题:转向语句视为有害)中指出,据彼观察,程序片的易读性与易理解性与其中所含之无条件转移控制(转向)个数成反比。该信发表后,引起普遍重视,结构化程序设计方法逐渐形成,并成为程序设计领域、计算机软件领域的重要方法,对计算机软件之发展意义重大,作用明显,相应出现了诸如 Modula-2, C, Ada 等所谓结构化程序设计语言。

结构化程序设计方法之主要贡献有二:一是将程序设计方法由技艺向科学迈进一步;二是据此方法所编出之结构化程序不只是供计算机阅读而且也是供人阅读的创造性工作。

参考文献

1. Bohm C, Jacopini G. Flow diagrams, turing machines, and languages with only two formation rules. Comm. ACM., 1964, 9(5)

2. Dijkstra E W. Go to statement considered harmful. Comm. ACM., 1968, 11(3)

3. Dahl O J, Dijkstra E W, Hoare C A R. Structured programming. New York: Academic Press, 1972

(徐家福 仲萃豪)

jiegouhua fangfa

结构化方法(structured method) 强调开发方法的结构合理性以及所开发软件的结构合理性的软件开发方法。结构是指系统内各组成要素之间的相互联系,相互作用的框架。结构化开发方法提出了一组提高软件结构合理性的准则,如分解与抽象、模块独立性、信息隐蔽等。针对软件生存周期各个不同的阶段,它有结构化分析(SA)、结构化设计(SD)和**结构化程序设计**(SP)等方法。

结构化分析方法给出一组帮助系统分析人员产生功能规约的原理与技术。它一般利用图形表达用户需求,使用的手段主要有**数据流图**(DFD)、**数据词典**(DD)、结构化语言、判定表及判定树等。数据流图以图形的方式表达目标系统中信息的变换和传递过程,有以下 4 个基本要素:数据流、加工、文件、数据源和数据宿。数据流由一组固定成分的数据组成,它具有名字和流向;加工是对数据流的变换;文件是可访问的存储信息;数据源和数据宿是存在于计算机系统之外的实体,分别表明数据处理过程的数据来源及数据去向。SA 方法采用分层的数据流图来表达一个目标数据处理系统,在保证一致性及完备性的情况下,采用数据流图分解及抽象的手段,来控制需求分析工作的复杂性。

与数据流图配合使用的是数据词典,它对数据流图中出现的所有数据元素给出逻辑定义,用以对数据流图中的各要素得到确切的解释。

在分层数据流图中,最底层的数据加工(称为基元)不需再用下层 DFD 进一步描述,而可采用结构化语言、判定表或判定树等描述其功能。结构化语言有基本的过程控制结构,如顺序、选择和重复,可对 DFD 中的功能进行过程化描述。

结构化分析的步骤如下:①分析当前情况,作出反映当前物理模型的 DFD;②推导出等价的逻辑模型的 DFD;③设计新的逻辑系统,生成数据词典和基元描述;④建立人机接口,提出可供选择的目标系统物理模型的 DFD;⑤确定各种方案的成本和风险等级,据此对各种方案进行分析;⑥选择一种

方案；⑦建立完整的需求规约。

结构化设计方法给出一组帮助设计人员在模块层次上区分设计质量的原理与技术。它通常与结构化分析方法衔接起来使用，以数据流图为基础得到软件的模块结构。SD 方法尤其适用于变换型结构和事务型结构的目标系统。在设计过程中，它从整个程序的结构出发，利用模块结构图表达程序模块之间的关系。

模块结构图是采用结构化设计方法进行软件概要设计的重要描述手段。它以图形的形式描述软件系统的模块组成及模块之间的调用关系。

构成模块结构图的主要成分有模块、调用和数据。在结构图中的模块以矩形表示，在矩形框内可以标以模块的名字。模块间如有箭头或直线相连，表明它们之间有调用关系。对于两个处在不同位置上的模块，通常把上面的称为调用模块，下面的称为被调用模块。调用箭头上的小箭头表示模块调用时模块间的数据传送，小箭头的方向指明了传送的方向。数据也应用适当的名字来标识。

结构化设计的步骤如下：①评审和细化数据流图；②确定数据流图的类型；③把数据流图映射到软件模块结构，设计出模块结构的上层；④基于数据流图逐步分解高层模块，设计中下层模块；⑤对软件模块结构进行优化，得到更为合理的软件结构；⑥描述模块接口。

参考文献

1. DeMarco T. Structured analysis and system specification. New York：Yourdon Press，1979

2. Yourdon E. Managing the system life cycle. New York：Yourdon Press，1982 （吕建国 钱乐秋）

jiegouhua fenxi yu sheji jishu

结构化分析与设计技术（structured analysis and design technique，SADT） 由一组称为结构分析图的图形语言工具与使用这组工具的方法、管理技术所构成的系统分析与设计。SADT 的结构分析图用于构成 SADT 模型来表达需求分析的结果。SADT 方法自顶向下地建立 SADT 模型。

SADT 模型由一组有序的结构分析图构成，每张结构分析图是由若干个结点以及连接这些结点的弧组成的工程图。SADT 模型是一种分层的系统模型，它自顶向下地把高层的结点分解成为若干个下层图。

SADT 模型有两种基本类型：以描述系统中的活动为主的活动模型和以描述系统中的实体为主的“数据模型”，分别用*活动图*和*数据图*表示。活动图的结点表示活动，弧表示活动之间的数据流；数据图表示结点内数据对象及其弧上的活动。因而，数据图和活动图是对偶的，活动图和数据图统称为结构分析图。一个完整的 SADT 模型是由多个视角的活动模型和数据模型构成。在实践中，活动图应用更广泛。

图 1 给出了活动图和数据图中结点的一般形式，每个结点有四种不同类型的弧，结点左边进来的弧是输入，结点右边引出的弧是输出，顶端进入的弧称为控制，而底部进入的弧称为机构。在图中，每个结点用四种类型的弧与其他结点发生联系。一个结点的输出必定是另一个结点的输入或控制，或者是整个系统的输出。每个结点的输入或控制也必定是另一个结点的输出或外部环境的输入。

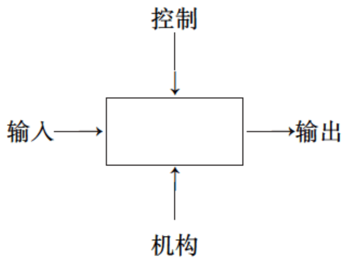


图 1　基本结点图

在活动图中，输入和控制表示完成此结点的活动所需要的数据，输出表示执行该活动所产生的结果数据，机构表示完成该活动的机构（人或设备）。在数据图中，输入是生成数据对象的活动，输出是使用数据对象的活动，控制是对结点活动条件的一种约束，机构是用来储存数据对象的设备。在活动图中的活动和数据图中的数据，均可进一步细化，形成一个层次的活动图或数据图，从而形成一个由多层次的结构分析图所构成的系统模型。

SADT 首先是由 T. D. Ross 在 Softech 公司从事大型软件开发的实践中，总结出来的一种通用的处理复杂问题的方法。1977 年由 Ross 等人发表的著名论文“需求定义的结构化分析”奠定了这一方法的基础。20 世纪 70 年代中后期，SADT 成为最有代表性的结构化软件开发方法之一。该方法被广泛地用于系统的定义、软件需求分析、系统设计以及软件设计，特别适于规模较大和复杂的系统的分析与设计，曾被 ITT 作为软件规约与设计的标准。这一方

法和技术中的活动图模型被美国空军公布的 ICAM 工程所采纳，并作为其 IDEF（ICAM definition method）软件开发方法的 IDEFO 模型，得到了广泛的应用和推广。（郝克刚　王斌君）

jieshi chengxu

解释程序（interpreter）　按照源程序中指令或语句的动态执行顺序，逐条翻译，并立即解释执行相应功能的处理系统。

解释程序的突出特点是不把源程序翻译成机器语言形式的目标程序，而是直接将源程序中的指令或语句转换成加工数据的动作或完成所需功能的动作。

解释程序是由总控程序和一组相应各种类型指令或语句的执行子程序组成（图 1）。其工作过程如下：首先，由总控程序完成初始化工作，将系统置成初态；然后，依次从源程序中取出一条指令或语句，并进行语法检查，如果语法有错，则输出错误信息；如果语法正确，则根据指令或语句的类型，转去执行相应类型的执行子程序；自执行子程序返回后，检查源程序是否解释完毕，如果未完成，则继续解释执行下一条指令或语句；如果完成，则由总控程序完成必要的善后处理工作。

解释程序的优点是实现算法简单，且易于在解释过程中灵活、方便地插入所需的修改和调试措施；

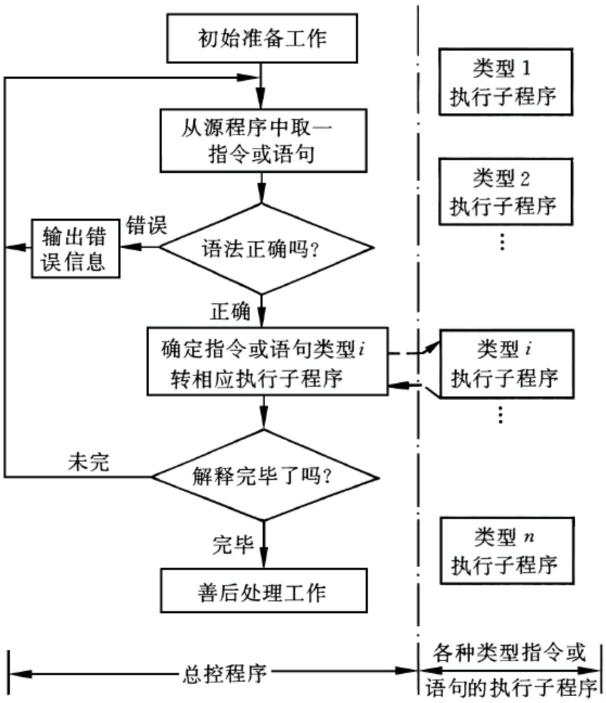


图 1　解释程序的结构和工作流程示意图

其缺点是运行效率低。例如，对于源程序中需要多次重复执行的指令或语句，解释程序将要反复地取出、翻译和解释执行它们。根据这些特点，解释程序通常适用于交互方式工作的，或调试状态下运行的，或运行时间与解释时间相差不大等类型的语言。随着超大规模集成电路的迅速发展，人们提出了诸如 FORTRAN 机、COBOL 机之类的高级语言机器的概念。其基本思想就是采用软件固化的方法，通过微程序设计语言实现高级语言的解释程序。伴随计算机硬件和软件技术的不断发展，解释程序将有广阔的应用前景和发展前途。（曹东启）

jieshi jizhi

解释机制（explanation mechanism）　指**专家系统**解释模块的功能原理。解释模块能提供专家系统行为的自我说明，即能对专家系统得出的结论（或做出的决策）给出推导过程和合理性解释。

解释机制能增加专家系统的透明性和其推出结论的可接受性，可提高用户对专家系统的信任程度。

解释模块一般能回答的问题有：

（1）推理状态检查　回答系统运行过程中常见的推理问题。包括回溯推理问题，假设推理问题和矛盾性推理问题。

（2）一般性问题回答　回答有关知识库中知识方面的问题。

解释机制不但能解释推理的路径，帮助用户理解**知识库**的内容，还具备一些辅助功能。诸如在系统维护时，可帮助知识工程师发现和更正知识库中的错误；在追踪推理路径时，可发现推理的不一致性；对初级用户的辅助教学作用：因专家系统的知识能有效支持高质量的决策，通过解释可形成一个教学过程，使用户学习权威专家的决策过程。

常用的解释机制包括预制文本法、执行追踪法、策略解释法和自动程序员方法。

预制文本法是将问题答案预先插入到程序中，通过显示这些文本来回答用户提问的方法。

执行追踪法通过对程序执行过程进行追踪，说明程序是如何得出结论的，即通过推理过程的重新构造来说明系统的动作。

策略解释法模拟人类推理，将控制知识用规则抽象地描述出来而不是隐含在具体代码中，并且与领域规则完全分开。

自动程序员法将自动程序设计的思想应用于解释机制，通过保留推理轨迹以及相关的信息，来记录

系统的动作。

早期专家系统大多缺乏附加的知识源(支持知识),因此解释能力基本局限于预置文本和程序代码的翻译(执行追踪)。预制文本法不能预测和适应用户的所有问题,执行追踪法也只能在较低的级别上解释系统的行为,无法对系统的高级行为给出合理的解释。更复杂的解释需要将隐藏在专家系统后的知识明确表示出来。策略解释法能够为用户提供一个策略上的解释。自动程序员法可向用户证明系统目前行为的合理性。

如何在推理效率和解释能力之间达到良好的平衡(即在保证系统推理效率的前提下尽可能为用户的所有问题提供满意的解释),是解释机制所要解决的主要问题。此外,现有解释机制还面临可回答问题范围窄、解释方式不够灵活、对用户持续提问无法给出回答、可扩展性差等局限性,这与知识库的构造方式和专家系统的生成策略有关。向专家系统中引入**自然语言生成**策略(natural language generation literature)和学习机制,构建逻辑一致、知识结构清晰的知识库有望解决上述问题。

参考文献

1. 管纪文, 刘大有,等. 知识工程原理. 长春: 吉林大学出版社, 1988

2. Rich E A. 人工智能引论. 李卫华,汤怡群,文中坚,译. 广州: 广东科学技术出版社,1986

3. Johanna D. Moore, William R. Swartout. Explanation in expert systems: A survey. Technical Report ISI/RR-88-228, Information Sciences Institute, USC, Marina del Ray, California, December 1988

(施鸿宝　黄晶)

jincheng

进程(process)　程序的一次执行过程。反过来看,程序则是进程的一种静态描述。进程是在操作系统的管理下被启动,进入运行状态,并在一定条件下中止或结束的。进程的运行需要使用一定的计算机资源(处理器、内存、各种外部设备等)。因此,进程又是操作系统调度的基本单位。

一个普通(非并发)的程序被执行时,相当于产生了一个新的进程;执行完毕后,这个进程就消亡了。为了提高程序的执行速度,可以把一个程序设计成多个可以"同时"执行的部分,这就是所谓的*并发程序*。执行并发程序时,可以有多个对应于此程序的进程同时处于运行状态。其中不同的进程如果在某一瞬间使用的是不同种类的资源,或者虽然是同一类,但是系统拥有多台这类设备(例如,多处理器系统中的多台处理器),那么这时这些进程是真正在物理上"同时"运行的。反之,它们就必须以某种次序轮番运行。但我们认为它们在逻辑上仍然是可以"同时运行"的。

然而,除了资源使用上的冲突之外,并发程序中的多个进程之间并非完全独立无关,它们之间往往需要相互合作,才能协同完成任务。例如进程 A 在运行到某一阶段后,必须等待进程 B 向它提交某种数据,然后才能继续运行,这就是一种通信关系。还有一种是"互斥关系",例如,有一个飞机订票系统,为每一订票终端都安排了一个对应的进程。每当有一个旅客在某一个窗口订票时,相应的进程便被启动,去查看是否有空位。如果有,就出售与该座位对应的票,并修改数据,以免该票被重复出售。但是,由于不同售票窗口对应的进程是可以"同时运行"的,因此有可能出现两个进程同时查看同一座位的情况:两个进程同时发现同一座位有空。于是仍然会出现一张票卖给两位旅客的错误。为了避免这一点,只需规定"查看是否有空位"的这一段程序在同一时刻最多只能让一个进程进入。

并行程序中的多个进程不一定是在一开始就同时出现并一直维持到整个程序结束,而是可以根据需要动态地产生和消亡。换言之,一个进程可以在运行过程中派生出新的进程来。新派生出来的进程又可以继续派生下一代的进程,于是形成了进程之间的"父子关系"。当一个新的子进程诞生时,它所需要的资源通常是由其父进程的资源中划分而来。不同的进程在互相独立的内存空间中运行,这一点是和"线程"不同的。同一进程内又可以分解为若干线程,这些线程在逻辑上也是可以同时运行的,但是它们共享同一内存空间,因此线程之间的调度无须操作系统的介入,从而减小了系统开销,提高了效率。　(周锡令)

jincheng daishu

进程代数(process algebra)　关于通信并发系统的代数理论的统称。

20 世纪 70 年代后期,英国学者 R. Milner 和 C. A. R. Hoare 分别提出了**通信系统演算**和**通信顺序进程**,开创了用代数方法研究通信并发系统的先河。此后这一研究方向兴盛不衰,出现了众多类似而又相互区别的演算系统,如 ACP(提出者 J. A. Bergstra

和 J. W. Klop)，ATP(提出者 M. Hennessy)，Meije(提出者 G. Boudol，R. de Simone)，LOTOS 等，统称为进程代数。这些代数理论都使用通信，而不是共享存储，作为进程之间相互作用的基本手段，表现出面向分布式系统的特征。

在语法上，进程代数用一组算子作为进程的构件。算子的语义通常用结构化操作语义方法定义，这样进程就可看成是带标号的变迁系统。进程代数的一个显著特征是把并发性归结为非确定性，将并发执行的进程的行为看成是各单个进程的行为的所有可能的交错合成，即所谓交错语义。

进程代数研究的核心问题是进程的等价性，即在什么意义下两个进程的行为相同？在进程代数领域使用的最为广泛的等价关系有互模拟、测试等价、失败等价(参见**通信顺序进程**)等。对这些语义等价关系均建立了相应的公理系统。关于公理系统的研究不仅加深了对语义理论的理解，而且使得有可能对语义等价关系进行形式推理。

为了将进程代数的理论成果应用于解决实际问题，20 世纪 80 年代后期出现了许多计算机支持工具。用这些工具可对进程的行为进行推理或模拟。

参考文献

Milner R. Operational and algebraic semantics of concurrent processes. Chapter 19 of Handbook of Theoretical Computer Science. van Leeuwen ed. Elsevier Science Publisher B. V.，1990

（林惠民）

jinhua jisuan

进化计算(evolutionary computation) 受到自然界的进化和自适应机制的启发而发展起来的一种计算模式。又名演化计算。进化计算的实现过程首先利用编码技术表示问题相关的复杂结构，再通过对编码表示进行遗传操作和自然选择来指导和确定搜索方向。在问题求解过程中，它通常会维持多个个体(一个种群)、并行搜索解空间内的多个区域，通过不断进化(迭代)的方式，逐步获得质量更优的解。因此，进化计算本质上也可视为一种复杂的试错法。

进化计算的起源可以回溯到 20 世纪 50 年代 Hans J. Bremermann、Richard M. Friedberg、George E. P. Box 等人的独立研究成果，他们试图将自然界的进化过程引入工程研究领域，以解决工程中的优化问题。60 年代起，由于 John H. Holland、Ingo Rechenberg、Hans-Paul Schwefel，以及 Lawrence J. Fogel 等多位学者在基础理论与算法设计等方面的巨大贡献，进化计算得到了根本性的发展。其中，Holland 于 60 年代初期提出**遗传算法**，并由 Kenneth A. De Jong、David J. Goldberg 等人进一步推广。与此同时，Rechenberg 提出按照自然突变和自然选择的生物进化思想，对物体的外形参数进行随机变化并进行优化，进化策略由此诞生。之后，Rechenberg 与 Schwefel 开展了一系列后续研究使得进化策略得到进一步完善。1966 年，Fogel 等人出版了《基于模拟进化的人工智能》一书，系统阐述了进化规划的思想，他们将给定问题描述成有限状态机，通过对其施加进化算子达到优化的目的。到了 80 年代，由于计算机计算能力的提高和并行计算机的兴起，进化计算对计算机性能的要求已不再是其发展的桎梏。同时，它的不断发展，尤其是在一些复杂应用问题上取得的成果，使得其受到人们越来越多的关注。

早期对进化计算的研究是由不同领域的研究者独立开展的，在相当长时期内他们之间并没有正式沟通。直到 90 年代，遗传算法、进化策略与进化规划的研究者才开始逐步接触并了解到对方的研究成果。他们发现彼此在研究中所依赖的基本思想都是基于遗传和自然选择等生物进化思想，具有惊人的相似之处，于是提出将这一类研究工作命名为“进化计算”，而将相应的算法统称为“进化算法”。

进化计算研究的主要任务在于借鉴自然界中的进化与自适应机制建立相应的计算模型，揭示计算模型的内在运行机制及其与待求解问题之间的关系，以面向现实世界的各种复杂难解问题，如 NP 难问题，设计有效的求解方法。其研究内容主要包括算法设计、算法的理论分析，以及算法应用研究等方面。

进化算法是借助于生物进化的原理发展而来的一类迭代算法，它从给定的初始解(个体)开始，通过对个体进行遗传操作、适应度评估，以及选择操作，逐步改进解的质量，并期望收敛到问题的最优解。问题编码机制、遗传算子，以及选择机制是进化算法的三个核心要素。其中遗传操作又主要包括交叉(重组)和变异两种。狭义地说，进化算法包括**遗传算法**、进化策略、进化规划和遗传编程四类类型。从广义上来说，很多基于某些自然现象或过程发展起来的算法，诸如群智能算法、人工免疫系统、差分进化、DNA 计算等，都可认为是属于进化计算的研究范畴。此外，在已有算法基础上提出的新的计算模型，如协同进化模型、并行/分布式进化算法等，也

是算法设计的重要研究内容。

遗传算法通常将问题的解编码成一组二进制串(称为染色体、个体)。然后采用基于适应度比例的选择机制在当前种群中选择个体,再使用交叉和变异两种操作来生成下一代种群,如此迭代进化下去,直到满足一定的终止条件。

和主要求解二值离散问题的遗传算法不同,进化规划算法是一种抽象的算法框架,对算法的编码和具体形态没有特别的要求,但经常被看作是一类用于数值优化问题的算法。进化规划算法框架区别于其他算法的显著特点在于它十分强调变异算子在进化过程中的作用,却不提倡在算法中使用交叉算子。最初的进化规划算法基于高斯分布进行变异操作。随后,其他类型的概率分布,如柯西分布、列维分布等,也被证明可用于设计有效的变异算子。

进化策略和进化规划类似,对算法的编码和具体形态没有特别的要求,并通常被用于求解连续空间上的数值优化问题,且主要基于高斯分布进行变异操作。但与进化规划不同,进化策略坚持在算法框架中保留交叉算子。

遗传编程是由 Nichael L. Cramer 于 1985 年提出,并由 John R. Koza 于 20 世纪 90 年代进一步完善的一种特殊的进化算法,用于计算机程序的优化设计与自动生成。它采用树状编码,这使得进化 LISP 或是汇编语言成为可能,而该算法中的遗传操作主要用于动态改变树状结构。

随着进化计算领域的不断发展,过去 20 年内涌现出了大量其他类型的进化算法,包括粒子群优化算法、蚁群算法、差分进化、人工免疫算法等。这些算法大都采用了与上述四种经典进化算法类似的框架,但并不直接起源于自然进化的思想,而是借鉴了自然界中其他形式的演变及自适应机制,并在此基础上采用了各具特色的遗传操作和选择机制。

几十年来,进化计算在工程优化、过程控制、经济预测以及模式识别等诸多领域取得了广泛的成功,正受到人们越来越多的关注。面向实际生产生活中更大规模、易收到多种不确定因素影响的复杂问题,研究效率更高的进化计算模型,是将进化计算进一步推向实用的必由之路。此外,虽然进化计算的灵感来源于自然法则,但其与许多更成熟的研究领域,如并行/分布式计算、随机算法、多智能体系统、运筹学等有着密切联系。脱离对自然现象的简单模仿,将进化计算的理论体系与这些领域已有的研究成果相对接,也是未来一个重要的研究趋势。

参考文献

1. 潘正君,康立山,陈毓屏. 演化计算. 北京:清华大学出版社,1998

2. 陈国良,王煦法,庄镇泉,王东生. 遗传算法及其应用. 北京:人民邮电出版社,1996

3. Eiben A E, Smith J E. Introduction to evolutionary computing. Berlin: Springer-Verlag, 2003

(姚新 唐珂)

jinsi suanfa

近似算法(approximation algorithm) 计算机科学中算法研究的一个重要方向。所谓“近似”,就是指结果不一定是最优的,但是也在可以承受的范围内,而且可以比精确求解消耗更少的资源。计算复杂性理论中,在 P≠NP 的假设下,已被证明为 NP-难解(NP-Hard)的优化问题,无法在多项式时间内求到最优解,然而在现实或理论研究中,不能回避这类问题的求解。在不能保证得到精确解的情况下,转而依靠高效算法寻求可以接受的近似最优解,是寻求近似算法的原始目的。因此近似算法作为一种算法,是寻求问题近似最优解的多项式时间算法,通常近似算法都有多项式时间复杂性。另外近似算法作为一个研究方向,以设计 NP-难解优化问题的近似算法为主要内容。近似最优解可理解为能被人们接受,但不保证是最优解的解。一种近似算法为绝对近似算法,要求算法求得的近似解与问题最优解数值之差的绝对值不超过某个常数。如最小度生成树问题,存储最多程序问题,都能设计算法求出与最优解数值相差不超过 1 的近似解。但在 P≠NP 假设下,多数 NP-难解优化问题都不存在这种绝对近似算法。因此另一种近似算法追求把算法求得的近似解与最优解数值之比值限定在某范围内,这种算法即通常所指的近似算法。为统一近似算法的衡量标准,对于最小优化问题,考虑“近似解/最优解”的数值为近似性能比;对于最大优化问题,考虑“最优解/近似解”的数值为近似性能比。近似性能比总不小于 1,越接近 1 说明算法能够求得的解越精确。一个算法的近似性能比为 B,是指算法求解问题的每个实例的近似性能比都不超过界值 B。如顶点覆盖问题存在近似性能比为 2 的近似算法,这个算法只需连续选择图中没有公共端点的边,将它的两个端点添加到一个点集 V 中,直到每条边至少有一个端点在 V 中。最后 V 中的顶点数目最多是覆盖所有边所需最少顶点数目的 2 倍。

再如背包问题,其实例为一个元素集合 A,和伴随每个元素的正整数对集合 $X=\{(P_i,W_i)|a_i\in A\}$,其中 P_i,W_i分别表示元素 a_i的价值和重量。要求寻找一个元素集合 A_1,使 A_1中元素重量不超过给定正整数 M,且总价值达到最大。只需要按照 P_i/W_i由大到小的顺序选择 A 中元素,直到选择下一元素则总重量超过 M 时停止,得到一个集合 A_2。再将 A_2中元素总价值与 A 中价值最大的元素价值相比较。若 A_2中元素总价值大则选择 $A_1=A_2$,否则选择 A_1只包含 A 中价值最大的元素。这样求出的 A_1一定满足:最优解价值 $/A_1$中元素价值≤ 2,及算法的近似性能比不超过 2。

一个问题能够设计出多么好的近似算法,是人们设计近似算法时存在的疑惑。问题本身决定了解答这个问题能够设计出多么好的近似算法。在 P≠NP 的假设下,大体可以把 NP-难解问题按照近似算法能够逼近最优解的程度分为 4 类。一类 NP-难解问题存在可任意逼近最优解的近似算法,即可以设定任意小的常数 ε,使算法有近似性能比为 $1+\varepsilon$,算法的时间复杂性是多项式的,当然其时间复杂性函数中必然含有常量“$1/\varepsilon$”,这样的近似算法称为多项式时间近似方案,如背包问题存在多项式时间近似方案。二类 NP-难解问题不存在多项式时间近似方案,但存在常数近似性能比的近似算法,如合取范式可满足问题、满足三角不等式的货郎问题都属于这一类。三类 NP-难解问题不存在常数近似性能比的近似算法,但存在近似性能比为 $O(\log n)$ 的近似算法,n 表示问题的输入数据规模,集合覆盖问题属于这一类。最后一类 NP-难解问题连 $O(\log n)$ 近似性能比的近似算法也不存在,但存在近似性能比为 n^{ε} 的近似算法,n 表示问题的输入数据规模,最大团问题、最大独立集问题都属于这一类。

近似算法并不一定局限于解答 NP-难解问题。有些问题是多项式时间可解答的,但有时为了加快运行速度,用近似算法求解也是一种选择。有些问题根本不知道是否为 NP-难解的,人们也用近似算法来解答这样的问题。如果一个算法含有随机计算步骤,算法每次运行得到的解可能有随机性,人们常用近似性能比的数学期望来表示随机算法的求解质量,常直接将这个近似性能比的数学期望称为随机算法的近似性能比。

参考文献

1. Hochba D S. Approximation algorithms for NP-hard problems. ACM SIGACT News, 1997, 28(2): 40-52

2. Garey M R, Johnson D S. Computers and intractability: a guide to the theory of NP-completeness. W H Freeman, 1979

3. 朱大铭,马绍汉. 算法设计与分析. 北京:高等教育出版社, 2009

(黄文奇 朱大铭)

jingjian zhilingji jisuanji

精简指令集计算机(reduced instruction set computer, RISC) 采用简化了的**指令系统**和**硬连线控制器**的计算机。RISC 是在高效的流水线技术(参见**计算机流水线**)的基础上充分利用指令并行执行和编译优化技术的计算机。

20 世纪 80 年代初,RISC 这一词刚刚问世时,它的含义是简化指令系统的计算机,它舍弃不常用的复杂指令,并充分改进频繁使用的寄存器操作基本指令的实际执行效率,把**微程序控制器**改为硬连线控制器,加强寄存器-寄存器操作指令,从而简化了计算机结构,提高了性能。后来,RISC 技术强调优化流水线设计,在这个基础上使大部基本指令的执行尽可能地在一个机器周期内完成。但商品化的 RISC 产品并不太追求指令系统和硬件结构的简化,而十分注重编译的优化,使硬件与软件设计密切结合,共同承担计算机性能提高的任务。

RISC 的发展历程

(1) RISC 的问世在 20 世纪 70 年代中后期,当时不少学者研究计算机指令系统的实际执行效率,他们通过统计和分析,得出著名的“20%: 80%”定律。该定律表明,在当时大部分计算机的指令系统中,只有约 20% 的指令是经常使用的,它们约占程序执行总指令数的 80%。而指令系统中的约 80% 的指令在实际上是很少使用的,它们一般只占程序执行总指令数的 20%。70 年代中后期,IBM 公司采用简化指令系统思想设计了 IBM 801,即对于常用的基本指令做优化设计,注重提高指令执行效率,而对于不常用的复杂指令,则尽量少用或不用,或靠例行子程序来完成。应该说,IBM 801 是 RISC 思想的最早实践。

超大规模集成电路(VLSI)工艺在 80 年代初取得了很大的发展,在处理机芯片上设置含有较多寄存器的通用寄存器堆已经很容易实现。这有利于使用寄存器-寄存器操作来实现大多数基本指令,如加法、减法和逻辑操作。操作数都取自芯片上的寄存器堆,以加速基本指令操作的执行,而且操作数地址

字段只需指明寄存器在堆中的编号，所以指令中地址字段也较短，有利于实现在一条 32 位字长的指令中包含 2 个源操作数地址和 1 个操作结果地址。这样便于实现固定格式和固定字长的指令字设计。80 年代初，美国加州大学 Berkeley 分校研制了单芯片的精简指令集计算机，定名为 UCB RISC Ⅰ和 UCB RISC Ⅱ，RISC 的名字正式问世。与此同时，斯坦福大学也研制出 RISC 类型的 MIPS 芯片。UCB RISC Ⅱ芯片使用了 3 万多个晶体管，MIPS 芯片使用了 25 万个晶体管，它们的运行速度都超过了当时著名的超级小型计算机 VAX-11/780，这在实践上证明了 RISC 思想的正确性。

（2）RISC 产业的形成　20 世纪 80 年代中后期，SUN 公司和 HP 公司使 RISC 真正成为工业产品，并且在市场上畅销。SUN 公司在 UCB RISC Ⅱ基础上设计了 RISC 产品 SPARC 微处理机，并且在其工作站中取代了 MC 68020；HP 公司也研制了 HP-PA 的 RISC 产品系列。当时，RISC 在从实验室走向工业生产的过程中，要解决的重要问题是保持应用软件的兼容性。SUN 公司和 HP 公司都在 UNIX 或类似 UNIX 的操作系统的基础上，在 **C 语言**源程序这一级上做到软件兼容；并花了很多力量发展 C 语言的编译优化技术，使 RISC 的流水线执行效率提高，从而进一步改善 RISC 工作站的性能。它们的 RISC 工作站的性能都大大超过了同时期的 CISC 工作站的性能。SUN 公司和 HP 公司在 RISC 技术上的成功，使得 80 年代末期几乎所有公司的工作站都采用了 RISC 技术，并采用 UNIX 或类似 UNIX 的操作系统。如 IBM 公司的 RS/6000，SGI 公司的基于 MIPS R2000，MIPS R3000 的工作站等。RISC 结构和 **UNIX 操作系统**成为工作站的标准平台。

在**微型计算机**领域内，在 80 年代中后期，Intel 公司的 CISC 产品 80286 和 80386 还是占绝对优势的，主要是因为在 80x86/MS-DOS 平台上开发的应用软件已有上亿个。为了保持与这么多的应用软件兼容，Intel 公司采取了逐渐向 RISC 技术过渡的策略。它在 80486 和 Pentium 中吸取了 RISC 的思想，即尽量使基本指令在一个机器周期内执行，并且减少复杂指令的执行周期数，同时仍保持 MS-DOS 与 Windows 环境下应用软件的兼容性。1993 年底，IBM 公司、Apple 公司与 Motorola 公司联合开发了 RISC 微处理机 Power 601 以及以后的 Power 60x，这使微型计算机走向 RISC 技术的总趋势更加明显。

RISC 的主要设计思想

经过多年的发展，RISC 实际上已成为一种处理机体系结构的设计思想。20 世纪 80 年代中期，RISC 结构的设计思想可归纳成 6 点：①大多数指令是单周期完成的；②采用 LOAD/STORE 结构；③硬连线控制器；④较少的指令数量和寻址方式；⑤固定的指令格式；⑥注重编译的优化。在 RISC 成为工业产品进入市场以后，RISC 思想有所发展，因为商品化的计算机指令系统中的指令数量不能太少，而且超大规模集成电路工艺的进展使得当时在单个芯片上集成几百万个晶体管是可以实现的。因此，商品化的 RISC 不太强调减少指令系统的指令数量和简化硬件结构，而是强调在流水线结构的基础上，使执行指令的平均周期数（CPI）尽量减少；或者说，使每个周期平均执行的指令数（IPC）尽量提高。由于流水线（参见**计算机流水线**）指令在执行过程中，存在着数据相关和转移相关问题，可以用编译优化指令调度技术来提高流水线执行效率。因此，减少 CPI 或提高 IPC 必须由硬件和软件共同来承担，以提高系统的整体性能。

另外，为了减少访问存储器的时间，RISC 处理机芯片内一般都包含超高速缓冲存储器（cache），而且数据 cache 和指令 cache 分开。

RISC 的发展趋势

RISC 的发展趋势是进一步提高指令执行的并行度。多发射结构可以提高并行度，但由于在每个机器周期内发射的多条指令之间可能存在着数据相关和转移相关问题，同时在相邻周期之间的指令也可能存在着相关性，使问题更为复杂，从而编译优化的任务也更为艰巨。

在硬件上用寄存器旁路和内部提前方法解决数据相关问题，相对比较容易。但有效解决转移相关仍较为困难，可采用转移预测部件，甚至设置专门的转移处理机。

RISC 的发展将着重于对指令流的处理和分档，争取在发射指令以前，依靠指令调度（有时还加上硬件）尽量解除指令之间的相关性，从而提高指令执行的并行性。在流水线结构设计中，除了传统的那些流水线功能以外，可以有专门从事指令调度和判别功能的机制。另外，在分发指令以前，就把指令流分成若干个线程，使之避免相关性，然后发向多执行部件或多流水线部件去执行，这种结构称为*多线程* RISC。

参考文献

1. 李三立,李亚民. RISC——单发射与多发射体系结构. 北京: 清华大学出版社,1994

2. Harolod S Stone. Introduction to Computer Architecture. SRA Press, 1979

（李三立）

jingxiang jihanshu shenjing wangluo

径向基函数神经网络(radial basis function neural network,RBFNN)　一种模拟人脑局部调整、相互覆盖接收域的典型前向**人工神经网络**模型。

径向基函数神经网络是一种三层前向神经网络,其拓扑结构如图1所示。第一层为输入层,由信号源节点构成,将网络与外界环境连结起来,节点数由输入信号的维数确定;第二层为隐含层(径向基层),其节点由径向基函数构成,实现输入空间到隐层空间的非线性变换;第三层为输出层(线性层),对输入模式做出响应,其节点由隐含层节点给出的基函数的线性组合来计算。由于径向基函数(RBF)网络是一种模拟了人脑中局部调整、相互覆盖接受域的神经网络结构,因此,它是一种局部逼近网络,现已证明它能以任意精度逼近任意连续函数。

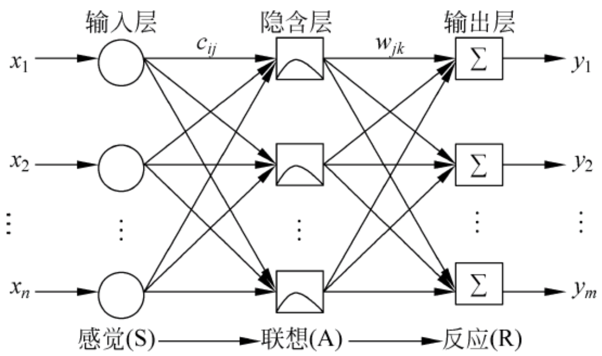


图1　RBF神经网络的结构

常用的径向基函数主要有高斯函数、多二次函数、逆二次函数和薄样条函数等,分别如式(1)~式(4)所示,其中最常用的径向基函数是高斯函数。

1. 高斯函数(Gaussian function)

$$\phi_i(x) = \exp\left(-\frac{\|x-c_i\|^2}{2\sigma_i^2}\right) \quad i=1,2,\cdots,h \tag{1}$$

2. 多二次函数(multiquadratic function)

$$\phi_i(x) = (\|x-c_i\|^2+\sigma^2)^{1/2} \quad i=1,2,\cdots,h \tag{2}$$

3. 逆二次函数(reciprocal multiquadratic function)

$$\phi_i(x) = (\|x-c_i\|^2+\sigma^2)^{-1/2} \quad i=1,2,\cdots,h \tag{3}$$

4. 薄样条函数(thin plate spine function)

$$\phi_i(x) = \|x-c_i\|^2\log(\|x-c_i\|) \quad i=1,2,\cdots,h \tag{4}$$

式中,σ_i 称为该基函数的扩展常数或宽度,σ_i 越小,径向基函数的宽度就越小,基函数就越具有选择性;c_i 为第 i 个径向基函数的中心;h 表示隐层神经元的个数。

径向基函数神经网络的学习算法分为两步:第一步是确定隐含层神经元数目、中心和宽度,第二步是确定隐含层和输出层之间的连接权值。径向基函数中心的选取方法主要有随机选取法、K-均值聚类算法、梯度训练方法和正交最小二乘法等。隐含层和输出层之间的连接权值的训练方法主要包括最小均方差、递推最小方差、扩展卡尔曼滤波等方法。

理论上已经证明,RBF神经网络具有良好的全局逼近特性。若RBF神经网络的隐含层神经元足够多,它可以在一个紧集上一致逼近任何连续函数。RBF神经网络是一种性能良好的前向网络,不仅具有最佳逼近性能,同时训练方法快速易行,不存在局部极小问题。这些优点给RBF神经网络的应用奠定了良好的基础,使其在函数逼近、模式识别和信号处理等领域都有广泛的应用。

参考文献

史忠植. 神经网络. 北京: 高等教育出版社,2009

（丁世飞）

jingtai suiji cunchuqi

静态随机存储器(static random access memory,SRAM)　（参见**半导体存储器**）

jingtai suiji cunqu cunchuqi xinpian

静态随机存取存储器芯片(static random access memory chip)　在芯片确定的地址范围内,可以对所选择的任一地址的存储单元写入和读出数据(随机存取),并在不断电时,不需要周期性刷新也可稳定保持所存数据的存储器芯片。但在断电时,静态随机存取存储器(static random access memory,SRAM)芯片所存数据会被破坏,因此它是一种易失性存储器芯片。

高速 SRAM 芯片按接口分,可分为非同步 SRAM 和同步 SRAM 芯片。非同步 SRAM 芯片是最早出现的 SRAM 芯片。同步 SRAM(SSRAM)是在高速 SRAM 基础上发展起来的以高速时钟同步并可以突发方式(burst mode)成组访问的 SRAM 芯片。其内核仍为非同步 SRAM,只增加时钟输入及相应变化的外围电路。进一步提高数据传输速率的技术是双倍数据速率(DDR)和四倍数据速率(QDR)的 SSRAM。按制造工艺分类,主要有双极型、NMOS 型、CMOS 型。20 世纪 80 年代后,由于互补金属氧化物-半导体(CMOS)工艺的成熟和极低的静态功耗,NMOS 和双极型 TTL 接口 SRAM 芯片已被 CMOS 型 SRAM 芯片代替。

在 SRAM 芯片中,存储一位数据(0/1)的基本存储单元由两个交叉耦合的反相器组成的触发器和两个由行选线控制的金属氧化物-半导体(MOS)管组成,如图 1 所示。行选线选中时为高电平,两 MOS 管通。触发器状态耦合到两条列线 BL 和 BL/,数据可读出。写入时,写入控制电路确定两列线的电平,通过行选线选中的 MOS 管改变触发器状态,将数据写入到存储单元。行选线未选中时为低电平,MOS 管截止,触发器状态不能传到列线,列线电平也不能改变触发器状态。只要不断电,数据可永久保存,不需周期性刷新。耦合到列线的信号幅度大,读出延迟小,读出时间快。但晶体管有 6 个之多,占用的芯片面积大,每个单元的平均价格高。

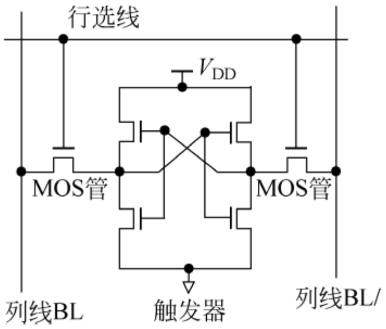


图 1 CMOS SRAM 芯片存储单元

SRAM 芯片的核心是存储单元阵列。芯片的存储容量专指存储单元阵列的单元总数。其大小由需存储的数据字数 2^N 乘以字长 M 位确定,N 为地址输入引脚数,M 为数据输入输出(I/O)引脚数(数据位数)。由于 M 远小于 2^N,为平衡阵列的行选择线和列线的负载,将输入地址分为行地址和列地址,行地址位数 N_R 位,列地址位数 N_C,即 $N = N_R + N_C$。存储单元阵列行选择线数(存储单元的行数)为 2^{N_R},列线有 $2^{N_C} \times M$ 对。通常 M 为 8/16/32/64,若需奇偶校验,则为 9/18/36/72。对应于每一位数据 I/O 线单元阵列为 2^{N_R}行 $\times 2^{N_C}$列,总存储容量等于 $2^N \times M$。图 2 为 4 Mb(256 K × 16 b)非同步 SRAM 芯片的构成框图。地址输入为 A0 ~ A17($N = 18$,$N_R = 11$,$N_C = 7$)。地址空间为 256 K 字。I/O 0 ~ I/O 15 为数据输入输出,即 $M = 16$。总容量为 4 Mb。

如图 2 所示,非同步 SRAM 芯片电路由存储数据的 SRAM 存储单元阵列、控制行选择线的行地址输入缓冲、译码和行选择线驱动电路、接收列线读出数据或向列线写入数据的读出放大和写入驱动电路、

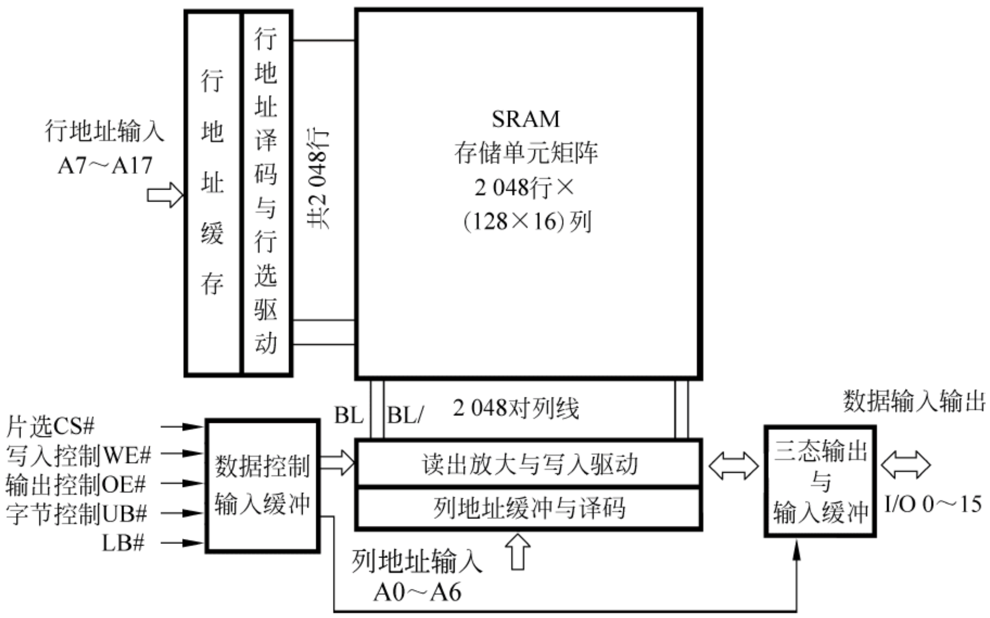


图 2 4 Mb(256 K × 16 b)非同步 SRAM 芯片的框图

控制列读写数据的列地址缓冲和译码电路、数据输入输出的接口电路和简单的数据流向控制电路组成。

非同步快速 SRAM 芯片读出时间(由输入确定的地址到输出该地址存储单元所存数据的时间),一般 4 Mb 芯片可达 8 ns(Vcc 为 3.3 V)。低功耗低速 SRAM 芯片的容量可达 32 Mb(2 M × 16 b),读出时间 55 ~ 85 ns。但目前已很少使用单独的非同步 SRAM 芯片。更多的是作为模块嵌入到系统或特殊的 SRAM 中使用,如双端口 SRAM、先进先出(FIFO)存储器。

为适应高速数字系统的要求,提高整个系统的性能,出现了同步 SRAM(SSRAM)芯片。SSRAM 芯片没有统一的 JEDEC 标准,其引脚和时序(表示其特点的名称)也不一定相同。SSRAM 工作方式有两个显著的特点:突发传输方式和流水线工作方式。由于同一行的存储单元在该行选中时,可将数据同时读出到列读出放大器和数据寄存器,实现同一行地址不同列地址数据的顺序读出或写入,即突发传输方式的读写操作。此时,数据和地址可按流水线方式重叠进行,提高了数据的有效传输率。

为了进一步提高数据传输速率,采用双倍数据速率(DDR)输入输出即 DDR SSRAM 芯片,此时输入时钟的上升和下降沿都是输入输出数据的参考沿。这样,可实现一个时钟周期读出或写入 2 个数据(对时钟的要求会有不同)。在成组传送时,数据的传送速率是芯片输入时钟频率的两倍。而四倍速(QDR) SSRAM 将数据输入端口和数据输出端口分开,读出数据和写入数据可分别以双倍数据速率从各自的端口输入和输出,利用流水线操作交替输入读出命令及地址和写入命令及地址。

SRAM 芯片的工作电压向低电压发展,以降低功耗。高速和高密度(大容量)工艺的发展也要求最高工作电压相应降低。电压值由初期的 5 V,逐步减为 3.3/2.5/1.8/1.5 V。芯片内核和 I/O 接口的工作电压可不同。

在现代计算机系统中,SRAM 主要用于实现高速缓冲存储器,包括嵌入在 CPU 芯片内的高速缓存(基于 SRAM 单元)和位于主板上的片外高速缓存(基于 SRAM 芯片)。通常高速缓存都采用关联存储器(即按内容编址存储器)结构,基于 SRAM 芯片的关联存储器也用于高性能的通信和网络设备,如在路由器中用于高速的路由查找和匹配。

参考文献

1. Sharma A K. 先进半导体存储器——结构、设计与应用. 曾莹,等译. 北京:电子工业出版社,2005

2. 桑野雅彦. 存储器 IC 的应用技巧. 王庆,译. 北京:科学出版社,2006

3. http://www.samsung.com/Products/Semiconductor

（孙祖希）

jingzhi tuxiang de yasuo bianma biaozhun

静止图像的压缩编码标准(compression and coding standards of still images)　参见**图像的压缩编码**。

jiucuo bianma

纠错编码(error correction code)　在数据传输过程中发生错误后能在收端自行纠正错误的编码。为使一种编码具有纠错能力,须对原码字增加多余的码元,以扩大码字之间的差别,即把原码字按某种规则变成有一定剩余度的码字,并使每个码字的码之间有一定的关系。码字到达收端后,可以根据编码规则是否满足以判定在数据传输过程中是否发生错误。当数据传输过程中发生错误时,按一定规则确定错误所在位置并予以纠正。纠错并恢复原码字的过程称为解码。检错码与其他手段结合使用,可以纠错。1948 年香农发表论文指出,只要采用适当的纠错码,就可在多类信道上传输消息。自香农的论文发表以来,人们经过持续不懈地努力已经找到多种好码,可以满足许多实用要求。但在理论上,仍存在一些问题未能解决。纠错码能够纠错,主要是靠码字之间有较大的差别。纠错码实现中最复杂的部分是解码,它是纠错码能否应用的关键。纠错码传输的都是数字信号,这既可用硬件实现,也可用软件实现。

分组码和卷积码是两类较重要的纠错码。分组码是对信源待发的信息序列进行分组(每组 K 位)编码,它的校验位仅同本组的信息位有关。自 20 世纪 50 年代分组码的理论获得发展以来,分组码在数字通信和数据存储系统中已被广泛应用。

海明码是一种可以纠正单个位差错的高效率的线性分组码,它利用在信息位为 m 位,增加 r 位冗余位,构成一个 $n = m + r$ 位的码字,然后用 r 个监督关系式产生的 r 个校正因子来区分无错和在码字中的 n 个不同位置的一位错。它必须满足以下关系式

$$2r \geqslant n + 1 \text{ 或 } 2r \geqslant m + r + 1$$

海明距离是海明码的一个基本概念，通常一帧包括 m 个数据位和 r 个冗余位或者校验位。设整个长度为 n（即 $n=m+r$），则此长度为 n 的单元通常被称作 n 位码字（codeword）。

给出任意两个码字（如 10001001 和 10110001），可以确定它们有多少个对应位不同。为了确定有多少位不同，只需对两个码字做异或（XOR）运算，然后计算结果中“1”的个数。计算方法如下

$$
\begin{array}{r}
10001001 \\
\text{XOR}\ 10110001 \\
\hline
00111000
\end{array}
$$

两个码字中不同位的个数，称为**海明距离**（Hamming distance）。其重要性在于，假如两个码字具有海明距离 d，则需要 d 个 1 位差错才能将其中一个码字转换成另一个。

一种编码的校验和纠错能力取决于它的海明距离。为检测出 d 比特错误，需要使用一个距离为 $d+1$ 的编码方案，因为在这样的编码方案中，d 个 1 位错误不可能将一个有效的码字转变成另一个有效的码字。当接收方看到无效的码字，它就能明白发生传输错误。同样，为了纠正 d 比特错误，必须使用距离为 $2d+1$ 的编码方案，因为在这样的编码方案中，合法码字之间的距离足够远，即使发生了 d 位变化，这个发生了变化的码字仍然比其他码字都接近原始码字。

卷积码不对信息序列进行分组编码，它的校验元不仅与当前的信息元有关，而且与以前有限时间段上的信息元有关。卷积码在编码方法上尚未找到像分组码那样有效的数学工具和系统的理论。但在译码方面，不论在理论上还是实用上都超过了分组码，因而在差错控制和数据压缩系统中得到广泛应用。

参考文献

1. 王达. 网络工程师必读——网络工程基础. 北京：电子工业出版社，2006

2. 胡道元. 计算机网络（高级）. 北京：清华大学出版社，1999

（周杰　张凌）

juyuwang

局域网（local area network）　位于局部区域内多台计算机等终端设备通过传输介质和通信设备互连起来的数据通信网络。所谓局部区域，可以是一个办公室、一栋大楼，或一个校园范围，通常覆盖距离小于几千米和几十千米。通信设备包括网桥、交换机和路由器等。通信传输介质分为有线和无线两种，如双绞线和微波。数据传输速率从早期的 1 Mb/s 和 10 Mb/s 逐步发展到 100 Mb/s、1 Gb/s、10 Gb/s 和 40/100 Gb/s 不等。在局域网内，通过高层协议和功能完善的网络软件，可以实现计算机等终端设备之间的相互通信和资源共享。局域网的典型特征为：①地理覆盖范围小（一般小于几十千米）；②传输速率高（大等于 1 Mb/s）；③传输误码率低（$10^{-8} \sim 10^{-11}$）。

按传输介质的通信方式来区分，局域网可分为基带传输和宽带传输两类。基带传输局域网在传输介质传输的是单一传输速率的数字信号，而宽带传输局域网在传输介质上传输的经高频调制的模拟信号，这是一种单向传输技术，同一介质可传输多个不同的频道，以支持数据通信和电视信号等。

常用的局域网有**以太网**、**令牌环网**、**令牌总线网**、**光纤分布式数据接口**（FDDI）、异步传送模式（ATM）、**快速以太网**和千兆、万兆、四万兆和十万兆位以太网等。常用局域网的主要技术特征如表 1 所示。

表 1　常用局域网的主要技术特征

拓扑结构	总线、环型、树型、星型
传输介质	同轴电缆、双绞线、光缆、微波、红外线、激光
数据传输速率（Mb/s）	4、10、16、100、1000、10 000、40 000、100 000
介质访问控制方法	CSMA/CD、令牌传递、CSMA/CA
介质传输技术	基带、宽带、IR、FHSS、DSSS

局域网技术包含三个主要方面：局域网拓扑结构、局域网介质访问控制方法和传输介质。

局域网拓扑结构　**局域网拓扑结构**是指用传输介质连接计算机等终端设备和通信设备的几何布局。目前常见的网络拓扑结构主要有以下五大类：总线型结构、星型结构、环型结构、树型结构以及星型和总线型相结合的复合型结构。

局域网介质访问控制方法　美国电气和电子工程师协会（IEEE）下属的 802 委员会对局域网制定了一系列协议标准，统称为 IEEE 802 局域网标准。它提出的**局域网基准（参考）模型**包括了**开放系统互连基准（参考）模型**的最低两层（物理层和数据链路层）的功能，其中数据链路层又划分为逻辑链路控

制子层和介质访问控制子层(参见**局域网逻辑链路控制子层**和**局域网介质访问控制子层**)。局域网介质访问控制方法是局域网介质访问控制子层的主要功能,即对物理传输介质进行访问管理。起初的局域网都是采用共享介质的通信技术,常见的有带碰撞检测的载波侦听多址访问-冲突检测(CSMA/CD)(参见**以太网**)、令牌传递等方法。共享介质通信技术要求发送站点首先向连接在同一介质上的所有站点广播其发送要求或标记,接着要确认是否被获准发送,然后再发送数据,这样可减少在介质上数据发送的碰撞。

传输介质　网络传输介质是指在网络中传输信息的载体,局域网的传输介质可分为有线和无线两类。常用的有线类传输介质主要有**双绞线**、**同轴电缆**和**光纤**,无线类传输介质主要有微波、红外线和激光等。不同的传输介质,其特性也各不相同,它们不同的特性对网络中数据通信质量和通信速度有较大影响。

参考文献

1. Alan chambers. IEEE standards for local and metropolitan area network. Overview and Architecture, 1999.3.16

2. 胡道元. 计算机局域网. 3 版. 北京:清华大学出版社, 2002

(黄令恭　张公忠　汪为农)

juyuwang jizhun (cankao) moxing

局域网基准(参考)模型(local area network reference model)　美国电气和电子工程师协会(IEEE)标准委员会根据国际标准组织(ISO)的**开放系统互连基准(参考)模型**,结合局域网自身的特点,于1980年2月提出的模型(IEEE 802基准(参考)模型)。局域网基准(参考)模型包含了OSI基准(参考)模型的物理层、数据链路层和其他更高层的部分内容,如图1所示。物理层与开放系统互连基准(参考)模型(OSI/BRM)的物理层相同,而开放系统互连基准(参考)模型的数据链路层被分为逻辑链路控制(LLC)和介质访问控制(MAC)两个子层,目的是要把数据链路层中涉及介质的部分和与介质无关的部分分开。

物理层为建立通信实体之间的信息传输物理连接,提供有关机械、电气、功能和规程等方面的特性,确保信号在通信信道上的正确传输。其功能主要包括提供传送数据的物理通路,二进制位信号的编码和解码,帧同步前导码的生成与去除,二进制位信号

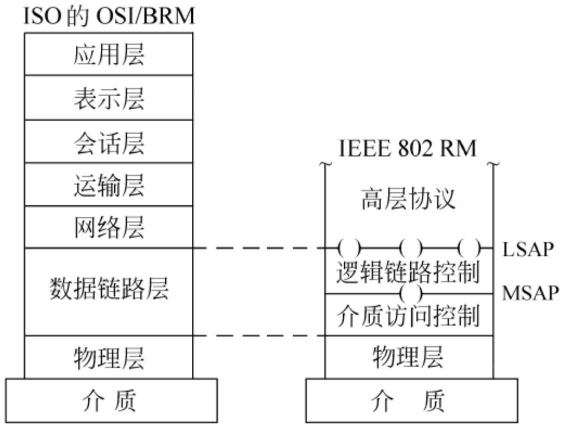


图1　IEEE 802基准(参考)模型

的发送与接收,错误校验(**CRC校验**)等。

介质访问控制子层(media access control, MAC)的主要功能是控制对传输介质的访问,为不同的物理介质定义了介质访问控制标准。MAC子层向逻辑链路控制子层提供一个介质服务访问点(MSAP)接口,在发送数据时,从上一层接收的数据组装成带MAC地址和差错检测字段的数据帧;在接收数据时拆帧,并完成地址识别和差错检测。

逻辑链路控制子层(logical link control, LLC)向高层提供一个或多个链路服务访问点(LSAP)接口,完成帧的接收和发送。它的主要功能包括数据链路的建立、维持和释放,为网络层提供面向连接的或无连接的服务;此外,LLC子层还提供寻址、差错控制、流量控制和发送顺序控制等功能,将不可靠的信道变为可靠的信道,确保数据帧的正确传输。

参考文献

Alan Chambers. IEEE standards for local and metropolitan area network. Overview and Architecture, 1999.3.16, http://www.ieee802.org/1/pages/802.html

(黄令恭　张尧学　汪为农)

juyuwang jiezhi fangwen kongzhi fangfa

局域网介质访问控制方法(medium access control method of local area network)　在局域网中对数据**传输介质**进行访问管理的方法。

局域网发展早期,对**以太网**(Ethernet)使用的共享介质(总线)进行访问控制的有带碰撞检测的载波侦听多址访问(CSMA/CD)、对令牌总线使用的共享介质或令牌环使用的介质或**光纤分布式数据接口**(FDDI)使用的介质有令牌(token)传递的访问控制方法。随着局域网的发展,以太网成了局域网的主

流，相应的带碰撞检测的载波侦听多址访问方法成了局域网的主流介质访问控制方法。随着以太网的规模不断扩大和数据流量的增长，这种带碰撞检测的载波侦听多址访问控制方法会引起严重的访问冲突，甚至网络阻塞。为解决此问题，出现了利用**网桥**、**交换机**将以太网的共享介质分隔成多个冲突域以减少冲突，形成了交换方式的介质访问控制方法。

CSMA/CD 是**以太网**采用的介质访问控制方法，它的基本原理是连接在以太网总线上的任何一台设备若有帧要发送，在任何时候都可以尝试发送这个帧，在发送前，先侦听总线的忙闲，发现总线空闲后，开始向总线发送这个帧；并在一个规定的时间间隔中不断检测总线上是否有其他设备也开始发送，即碰撞检测。若有碰撞，各方放弃此次尝试，并随机等待一段时间再尝试，直至发送成功。

*令牌传递*是令牌总线或令牌环采用的介质访问控制方法。*令牌*是一个专用的控制帧，它不停地依次在各个站点间传递，一个站点收到并占有令牌帧才可以发送。若某个站点有数据帧要发送，它必须等待令牌帧的到来，一旦它收到了令牌帧，不将此令牌帧转发给下一个站点，即占有该令牌帧，而是发送自己的数据帧。数据帧发送完毕后，该站点将令牌帧发往下一个站点，即释放令牌帧，以便其他站点占用和发送数据。

*交换方式*是解决共享介质的访问冲突，进一步提高网络有效带宽的另一种以太网介质访问控制方法。在一个共享介质上连接的站点越多，每个站点得到的平均带宽就越少，而且，随着站点增多，访问冲突加剧，造成有效带宽更为减小。解决此问题的一个方法是用交换机把一个共享介质分隔成多个网段，把一个冲突域分隔成多个冲突域。减少每一网段上的站点数，就增加了每个站点享用的平均带宽。如网段上只有一个站点，站点就可以享用全部带宽。站点数的减少，降低了访问冲突，也提高了有效带宽。

交换机在**局域网**中处在相当于**集线器**的位置，但不像集线器那样向所有端口转发输入帧，而是检查输入帧的目的地址，按目的地址将输入帧转发到相应的端口。

参考文献

1. Christensen K J, Haas L C, Noel F E, et al. Local area network-evolving from shared to switched access. IBM Systems Journal, 1995, 34(3)

2. http://en.wikipedia.org/wiki/Media_Access_Control

（黄令恭　王能）

juyuwang jiezhi fangwen kongzhi ziceng

局域网介质访问控制子层（medium access control sublayer of local area network）　**局域网基准（参考）模型**中的一个子层。介质访问控制（MAC）子层通过一个服务访问点（MSAP）向**逻辑链路控制（LLC）子层**（参见**局域网逻辑链路控制子层**）提供数据传送服务，完成数据链路层中的成帧、拆帧和**流量控制**以及对**局域网**数据传输介质的正确使用等功能。介质访问控制（MAC）是指局域网（LAN）各站点对**共享传输介质**可使用的权利和应遵守的规则。

MAC 子层的主要功能是：①帧的定界和识别；②目的站点的寻址；③源站点寻址信息的传送；④逻辑链路子层协议数据单元（PDU）的透明传送；⑤差错检测；⑥对物理传输介质的访问管理。

对应不同的**局域网网络拓扑结构**和传输介质，局域网介质访问控制采用不同的方法。常见的有：①用于总线型、星型或树型结构，采用**双绞线**或**光纤**为介质的 CSMA/CD 方法；②用于环型结构，采用双绞线或光纤的令牌传递方法；③用于**无线局域网**的 CSMA/CA 方法。

当 LAN 上用户站点数增多时，其业务量也随之增加，可能引起通信性能的下降，这是共享介质访问的 LAN 共同存在的问题。解决的方法是用**交换机**（参见**交换机**）将之分段，以减少同一个**局域网**上的用户站点数和业务量。

参考文献

Stallings W. Local and metropolitan area networks. 6th ed. Enlgewood Cliffs, NJ: Prentice Hall, 2004

（黄令恭　汪为农）

juyuwang luoji lianlu kongzhi ziceng

局域网逻辑链路控制子层（local link control sublayer of local area network）　**局域网基准（参考）模型**中的一个子层（LLC），它的主要功能是：建立和释放数据链路层的逻辑连接，通过一个或多个服务访问点（LSAP）向上层提供**无连接**和**面向连接**的数据传送服务，并利用介质访问控制子层（MAC）（参见**局域网介质访问控制子层**）去适应采用不同类型介质局域网的工作。

局域网协议标准（IEEE 802）为局域网逻辑链

路控制子层定义了三种逻辑链路控制服务类型：

（1）类型 1　不确认的无连接服务。数据帧在 LLC **实体**之间交换，无须**同等实体**之间事先建立逻辑链路。对这种数据帧的交换既不提供接收确认操作，也无任何流量控制和差错恢复功能。

（2）类型 2　面向连接服务。任何数据帧在交换前，首先需在一对 LLC 实体之间建立一条逻辑链路。数据传送时，数据帧依次发送，并提供差错恢复和流量控制功能。

（3）类型 3　有确认的无连接服务。数据帧交换之前无须先建立连接，但数据帧的发送和接收必须经过确认。这种服务常用于传送某些非常重要且时间性要求很强的数据。

LLC 子层把发往（或来自）高层的分组封装成 LLC 帧，即把分组作为 LLC 帧的数据字段，加上目的服务访问点（DSAP）字段和源服务访问点（SSAP）字段以及一个控制字段构成一个 LLC 帧（图 1）。这个 LLC 帧在经过介质访问控制（MAC）子层时还要加上 MAC 数据字段以形成 MAC 帧。LLC 帧中的控制字段的格式是参照**高级数据链路控制规程**（HDLC）中的平衡模式制定的，如服务类型 2 的 LLC 帧控制字段把 LLC 帧分为信息帧、监控帧和无编号帧，其中信息帧用来传送数据信息，监控帧用于确认、流量控制和差错控制，无编号帧用于连接建立、连接终止及其他控制功能。

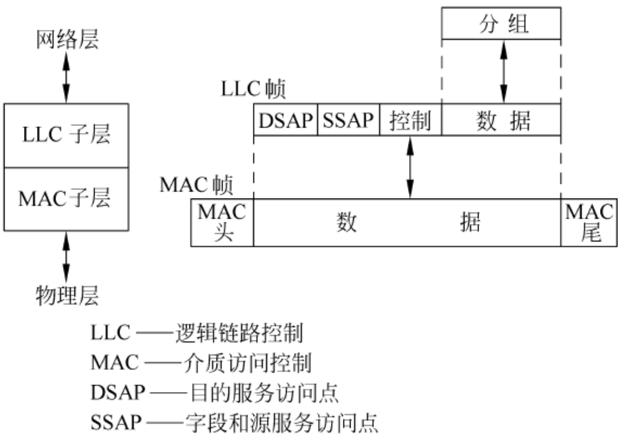


图 1　LLC 帧格式

局域网协议标准（IEEE 802）还涉及逻辑链路控制子层管理和安全方面的内容，LLC 子层中的“LAN 管理”模块提供在 LAN 站点之间交换管理信息用的数据链路层管理协议，并为所有局域网协议标准规定了被管理的对象；LLC 子层中的安全数据交换（SDE）实体，直接在介质访问控制子层上面提供安全的无连接服务。

参考文献

1. Stallings W. Local and metropolitan area networks. 6th ed. Englewood Cliffs, NJ: Prentice Hall, 2000

2. 张尧学，等. 计算机网络与 Internet 教程. 北京：清华大学出版社，1999　（黄令恭　汪为农）

juyuwang tuopu jiegou

局域网拓扑结构（topology of local area network）　指在**局域网**中网络节点（计算机等终端设备和通信设备）及通信线路（传输介质）的几何布局，即网络中节点及其通信线路的互连模式。基本的局域网拓扑结构有：全互连型结构、总线型结构、星型结构、环型结构和树型结构等，当前以星型和树型结构最为常见。

全互连结构　如果一个网络只连接几台设备，最简单的方法是将每一台设备都与其他设备直接互连，用这种方式互连形成的网络称为全互连网络，如图 1 所示。采用这种结构的网络通信的可靠性最高，但所需要的通信线路和接口设备也最多，因此这种结构的网络通常用在通信可靠性要求较高，网络规模较小的场合。

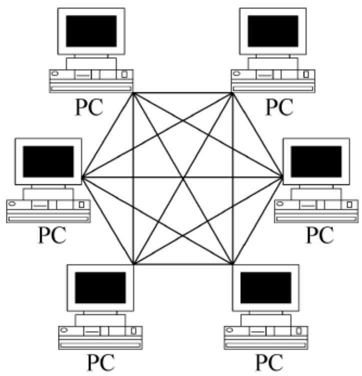


图 1　全互连型结构

总线型结构　使用同一**传输介质**连接所有节点的一种方式，即连接节点的物理介质被所有设备共享，如图 2 所示。任何一个节点发送的信号都可以沿着传输介质传播，而且能被其他所有节点接收。总线型拓扑结构的优点是：结构简单，因此网络可靠性较高，由于设备投入量少，组网成本也较低；网络可扩展性好，节点的增加或拆除不影响其他节点间的通信，因此易于维护。缺点是由于所有节点通信均通过一条共享总线，一次仅能一个节点发送数据，

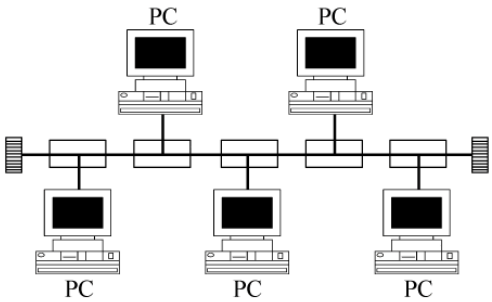


图 2　总线型结构

其他节点必须等待获得发送权，随着总线上节点数量的增多，节点间通信速度会下降，网络的规模也因此受到一定限制；由于共享总线，节点发送信息时比较容易发生冲突，故这种结构的网络通信实时性不强，网络效率和传输性能也不高。

星型结构　由一个中心节点和多个外围节点组成，如图 3 所示，外围节点通过点到点链路接到中心节点，中心节点是网络中唯一的转发节点，任意两个外围节点间的通信必须通过中心节点转发。通常中心节点由**交换机**或集线器（hub）组成。其优点是：网络结构简单，建网容易，便于控制和管理；任意两个外围节点间的通信最多只需两步，所以通信速度快；单个外围节点故障不会影响其他节点间的通信，便于维护。但缺点是一旦中心节点出现故障将导致全网瘫痪，因此网络可靠性差。

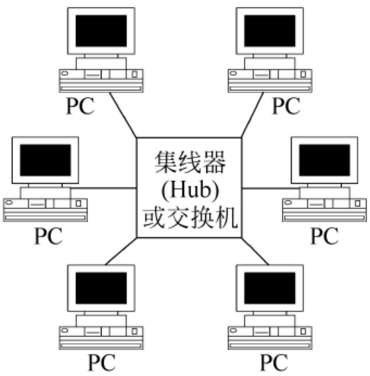


图 3　星型结构

环型结构　网络接口设备和传输介质串接成一个闭合的环路，每个入网设备通过网络接口设备接入环路（见图 4）。在环型网络中，信息流一般是单向的，每个节点都向它的下游节点转发它收到的信息包，信息包在环网中环绕一圈，最后由发送节点回收。由于多个节点共享一个环，就需要一种介质访问控制方法来决定每个节点何时发送数据。它的优

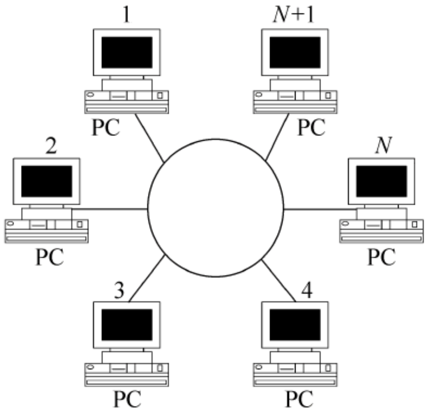


图 4　环型结构

点是它提供的对介质访问的灵活控制；此外，网络结构简单，投资少。其缺点是如果环的某一点断开，环上所有节点间的通信便会终止，可靠性较差；由于环路是封闭的，增加节点必须断开环路，因此可扩展性也较差。环型网络的一个典型代表是**令牌环网**。

树型结构　实际上是星型结构的扩展，通过多级星型结构的级连形成，如图 5 所示。树型结构中，叶节点通常代表计算机或其他终端设备，非叶节点一般由交换机或集线（Hub）组成，它采用分级的集中控制方式，其传输介质可有多条分支，但不形成闭合回路，每条通信线路都支持双向数据传输。树型结构除了具有星型结构的所有优点外，还具有以下自身优点：扩展性好，网络中节点扩充方便、灵活；结构比较简单，网络成本低，易于网络维护。缺点是对根节点依赖性大，如果根节点发生故障，则全网不能正常工作；同时，在级数较多的情况下，数据要经过多级传递，传输时间较长。

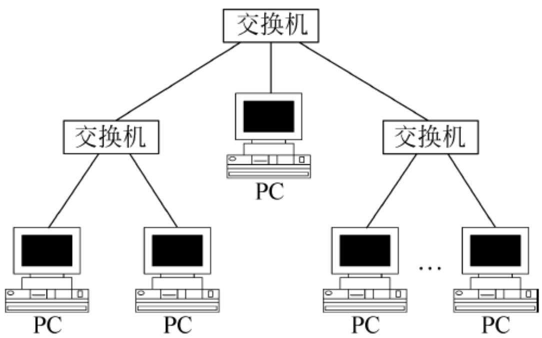


图 5　树型机构

参考文献

肖代华. 几种常见的局域网拓扑结构. 电脑报, 1998,（11）

（张博锋　汪为农）

juzhen jisuan

矩阵计算（matrix computation）　矩阵的加法、减法、数乘以及求逆等运算的总称。mn 个数 a_{ij} $(i=1,2,\cdots,m;j=1,2,\cdots,n)$ 的矩形阵

$$A=\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \tag{1}$$

称为 m 行 n 列矩阵或 $m\times n$ 阵，简记为 $A=(a_{ij})_{m\times n}$，a_{ij}称为 A 的元素。如果 $m=1$，称 A 为行矩阵；如果 $n=1$，称 A 为列矩阵；如果 $m=n$，称 A 为方阵，n 为 A 的阶数。当 $i\neq j$ 时，$a_{ij}=0$，称 A 为对角矩阵。当对角矩阵对角线上的元素全为 1 时，称为单位矩阵，常用 I（或 E）表示。

矩阵的相等、加、减、乘、转置与共轭　两个阶数相等的矩阵 $A=(a_{ij})_{mn}$，$B=(b_{ij})_{mn}$，$A=B$ 是指它们的对应元素相等（$a_{ij}=b_{ij}$）；$C=A+B$ 是指 A 和 B 对应元素相加（$c_{ij}=a_{ij}+b_{ij}$）所形成的矩阵。矩阵加法满足结合律和交换律。矩阵 $-A=(-a_{ij})_{mn}$ 称为 A 的负矩阵，$A+(-A)=0$。两个 $m\times n$ 矩阵的减法是 $A-B=A+(-B)$。

如果 A，B 分别是 $m\times n$ 与 $n\times q$ 矩阵，称 B 对于 A 是可乘的，其乘积 $C=AB$ 是一个 $m\times q$ 矩阵，它在 (i,j) 位置上的元素为 $c_{ij}=\sum_{k=1}^{n}a_{ik}b_{kj}(i=1,2,\cdots,m;\ j=1,2,\cdots,q)$。假如 A，B 是同阶方阵，那么 AB 与 BA 都有定义。与数的情况不同，这里可能出现 $AB\neq BA$ 以及当 $A\neq 0$ 及 $B\neq 0$ 时，$AB=0$ 的情况。矩阵乘法满足结合律和分配律。

$m\times n$ 矩阵 A 的转置矩阵是一个 $n\times m$ 矩阵，记为A^{T} 或 A'。对所有 i，j，A^{T} 的第 i 行是 A 的第 i 列，A^{T} 的第 j 列是 A 的第 j 行。如果 B 对于 A 是可乘的，那么有 $(AB)^{\mathrm{T}}=B^{\mathrm{T}}A^{\mathrm{T}}$。如果 $A=A^{\mathrm{T}}$，称 A 是对称矩阵；如果 $A=-A^{\mathrm{T}}$，称 A 是反对称矩阵。显然，对称矩阵与反对称矩阵一定是方阵。如果有 $A^{\mathrm{T}}A=AA^{\mathrm{T}}=I$，称 A 是正交矩阵。矩阵 A 的共轭矩阵 $\bar{A}$ 是 $\bar{A}=(\bar{a}_{ij})_{mn}$，这里$\bar{a}_{ij}$是 a_{ij}的共轭复数。如果 $\bar{A}=A^{\mathrm{T}}$，则称 A 为厄米特矩阵。

快速矩阵乘法　设 A，B 是 n 阶方阵，根据矩阵乘法定义，计算 $C=AB$ 的运算量是 $O(n^3)$。1969 年，V. Strassen 提出了运算量为 $O(n^{2.8074})$ 的**快速矩阵乘法**。该方法的基本思想是将 2 阶矩阵乘法所需的乘法数从 8 个减为 7 个，对于阶为 2 的乘幂的矩阵分解为 2×2 的矩阵块，块中元素均为 $n/2$ 阶方阵，一直递归分解下去。经 $\log_2 n-1$ 步后转化为 2 阶矩阵。

行列式　设矩阵 $A=(a_{ij})$ 是 n 阶方阵，其行列式定义为一个数，用

$$|A|=\begin{vmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{vmatrix}$$

或 det A 表示，它是 A 的所有元素的函数，其值是所有符合以下条件项的代数和。

- 每项是 A 中每行、每列各取一个元素组成的 n 个元素的乘积，可记为
$$a_{1j_1}a_{2j_2}\cdots a_{nj_n},$$
- 其中 $j_1,j_2,\cdots,j_n$ 是 $1,2,\cdots,n$ 的一个重排。
- 每项 $a_{1j_1}a_{2j_2}\cdots a_{nj_n}$前带正负号，以 1，2，…，$n$ 的顺序为标准来比较排序 $(j_1j_2\cdots j_n)$ 的逆序数是奇次或偶次而决定：奇带负号，偶带正号。

行列式的计算常用高斯消去法来完成。令

$$a_{ij}^{(1)}=a_{ij},\quad i,j=1,2,\cdots,n$$

则逐次消去过程为

$$a_{ij}^{(k+1)}=a_{ij}^{(k)}-a_{ik}^{(k)}a_{kj}^{(k)}/a_{kk}^{(k)},\quad k=1,2,\cdots,n-1;\quad i,j=k+1,k+2,\cdots,n$$

此时

$$|A|=\prod_{k=1}^{n}a_{kk}^{(k)}$$

在消去过程中要用 $a_{kk}^{(k)}$ 作除数，故要求 $a_{kk}^{(k)}\neq 0$。同时为了减少误差的积累，应取大元素作除数，这可用选主元的办法解决。例如采用行主元格式，即取 $a_{il}^{(k)}=\max_{j\geqslant k}\{|a_{ij}^{(k)}|\}$。如 $l\neq k$，则将第 k 列和第 l 列对调，同时将行列式变号。如 $a_{il}^{(k)}=0$，则 $|A|=0$。用高斯消去法求行列式的值所需运算量为 $O(n^3)$。如果应用舒尔余子式，还可建立运算量为 $O(n^{2.8074})$ 的行列式快速求值方法。

矩阵求逆　如果 n 阶方阵 A 的行列式不为零，那么就称 A 为满秩矩阵（或非奇异矩阵，或可逆矩阵），否则就称 A 为亏秩矩阵（或奇异矩阵，或不可逆矩阵）。一个满秩矩阵 A 有唯一的逆矩阵A^{-1}，使得$A^{-1}A=AA^{-1}=I$。通常矩阵求逆通过消元法完成。在消元法中，每步可以将对角线以外的元素消为零，并使得对角线元素为 1，从而得到

$$N_nN_{n-1}\cdots N_1A=I$$

其中N_k 为在第 k 步将矩阵$A^{(k)}=N_{k-1}\cdots N_1A$ 的第 k

列变为 e_k(这里 e_k 是第 k 个分量为1，其余分量为0的列向量)的矩阵。

求逆的具体步骤如下进行。设 a_{ij}，a_{ik}，a_{kj}，a_{kk} 均指当时在 $\boldsymbol{A}$ 的相应元素位置上的数，对于 $k=1,2,\cdots,n$ 的每个 k，依次做

(1) 计算 $d=1/a_{kk}$，并放在 a_{kk}处；

(2) 对于 $i\neq k$ 计算 $-da_{ik}$，并放在 a_{ik}处；

(3) 当 i 和 j 均不等于 k 时，计算 $a_{ij}+a_{ik}a_{kj}$，并放在 a_{ij}处；

(4) 对 $j\neq k$ 计算 da_{kj}，并放在 a_{kj}处。

最后在矩阵原来的位置上就得到 $\boldsymbol{A}$ 的逆矩阵$\boldsymbol{A}^{-1}$，所需的运算量为 $O(n^3)$。

矩阵求逆除了上述消元法之外，还有加边法和迭代法等。另外，通过矩阵的分块以及快速矩阵乘法可以得到快速矩阵求逆算法。

三对角矩阵求逆 在 n 阶矩阵 $\boldsymbol{A}=(a_{ij})$ 中，如果 $j-i\geqslant 2$ 或 $i-j\geqslant 2$ 时，有 $a_{ij}=0$(其中 $1\leqslant i,j\leqslant n$)，则称 $\boldsymbol{A}$ 为三对角矩阵。三对角矩阵的求逆可按下列步骤进行：设 $l_0=1$，$c_n=a_{nn}$，对于 $i=1,2,\cdots,n$，令

$$l_i=\begin{vmatrix} a_{11} & a_{12} & & \\ a_{21} & a_{22} & \ddots & \\ & \cdots & \cdots & a_{i-1,i} \\ & & a_{i,i-1} & a_{ii}\end{vmatrix}$$

$$c_i=\begin{vmatrix} a_{ii} & a_{i,i+1} & & \\ a_{i+1,i} & a_{i+1,i+1} & \ddots & \\ & \cdots & \cdots & a_{n-1,n} \\ & & a_{n,n-1} & a_{nn}\end{vmatrix}$$

另外，记

$$u_{i-1}=a_{i,i-1}a_{i-1,i},\quad 2\leqslant i\leqslant n$$

递归地作

$$l_i=a_{ii}l_{i-1}-u_{i-1}l_{i-2},\quad i=2,3,\cdots,n$$

计算

$$b_{ij}=\begin{cases}\left((-1)^{i+j}c_{i+1}l_{j-1}\prod\limits_{m=j+1}^{i}a_{m,m-1}\right)l_n^{-1}\ (j\leqslant i)\\ \left((-1)^{i+j}c_{j+1}l_{i-1}\prod\limits_{m=i}^{j-1}a_{m-1,m}\right)l_n^{-1}\ (j>i)\end{cases}$$

则有

$$\boldsymbol{A}^{-1}=(b_{ij})$$

上述三对角矩阵求逆算法的运算量为 $O(n^2)$。

参考文献

1. 冯康，等. 数值计算方法. 北京：国防工业出版社，1978

2. 游兆永. 线性代数与多项式的快速算法. 上海：上海科学技术出版社，1980

3. Golub G H, van Loan C F. 矩阵计算. 3版. 袁亚湘，等译. 北京：人民邮电出版社，2011

(蒋增荣 成礼智 安恒斌)

juzhen tezhengzhi wenti shuzhi jiefa

矩阵特征值问题数值解法(numerical solution of matrix eigenvalue problems) 在计算机上求解矩阵特征值及相应特征向量的数值方法。对于 n 阶矩阵 $\boldsymbol{A}$，求数 λ 和非零向量 $\boldsymbol{x}$ 使 $\boldsymbol{Ax}=\lambda\boldsymbol{x}$，称为矩阵特征值问题，其中 λ 和 $\boldsymbol{x}$ 分别是矩阵 $\boldsymbol{A}$ 的特征值和特征向量。矩阵特征值问题数值解法主要包括三类：QR 方法、幂法和子空间方法。

QR 方法 目前计算中小规模矩阵全部特征值和特征向量的最有效方法。它的理论基础是 Schur 分解定理：对于一个 n 阶矩阵 $\boldsymbol{A}$，存在一个酉矩阵 $\boldsymbol{Q}$ 和上三角矩阵 $\boldsymbol{R}$，满足 $\boldsymbol{AQ}=\boldsymbol{QR}$。容易看出，$\boldsymbol{R}$ 的对角元就是 $\boldsymbol{A}$ 的特征值。QR 方法通过一系列酉相似变换将给定矩阵逐步约化为上三角矩阵或拟上三角矩阵，从而得到近似的 Schur 分解。对于矩阵 $\boldsymbol{A}$，QR 方法的基本迭代格式如下：$\boldsymbol{A}_m-1=\boldsymbol{Q}_m\boldsymbol{R}_m$，$\boldsymbol{A}_m=\boldsymbol{R}_m\boldsymbol{Q}_m(m=1,2,\cdots)$；其中 $\boldsymbol{A}_0=\boldsymbol{A}$，$\boldsymbol{Q}_m$ 为酉矩阵，$\boldsymbol{R}_m$ 为上三角矩阵。在适当条件下，$\boldsymbol{A}_m$ 所有或大部分的对角线以下元素将趋于0。对于一般的 n 阶矩阵，一次 QR 迭代的计算复杂度是 $O(n^3)$。因此，在实际计算中，先将原矩阵 $\boldsymbol{A}$ 经一系列相似变换约化为一个上 Hessenberg 矩阵 $\boldsymbol{H}$，然后再对 $\boldsymbol{H}$ 进行 QR 迭代。上 Hessenberg 矩阵的 QR 分解一般利用 Householder 变换或 Givens 变换来实现，并不显示生成矩阵 $\boldsymbol{Q}$，其计算复杂度是 $\boldsymbol{O}(\boldsymbol{n}^2)$。为加速 QR 迭代的收敛，对于给定的上 Hessenberg 矩阵 $\boldsymbol{H}$，构造带原点位移的 QR 迭代：$\boldsymbol{H}_m-\mu_m\boldsymbol{I}=\boldsymbol{Q}_m\boldsymbol{R}_m$，$\boldsymbol{H}_m+1=\boldsymbol{R}_m\boldsymbol{Q}_m+\mu_m\boldsymbol{I}$；其中 $\boldsymbol{H}_0=\boldsymbol{H}$，位移量 μ_m 通常取 $\boldsymbol{H}_m$ 的右下角元素 $h_{nn}{}^{(m)}$，或右下角二阶子矩阵的特征值中最接近 $h_{nn}{}^{(m)}$ 者。带原点位移的 QR 算法的渐近收敛速度至少是2阶，对于 Hermite 矩阵时可达3阶。在求出近似特征值之后，可采用反幂法计算近似的特征向量。通用的线性代数软件包 LAPACK(Linear Algebra PACKage)提供的矩阵特征值问题数值解法器正是基于 QR 方法。

幂法 是计算矩阵模最大的特征值及相应特征向量的一种迭代方法。对于给定矩阵 $\boldsymbol{A}$ 和标准化

(归一化)初始向量$\boldsymbol{v}_0$,幂法的迭代格式如下:$\boldsymbol{v}_m = \boldsymbol{A}\boldsymbol{v}_{m-1}$, $\theta_m = \boldsymbol{v}_{m-1}{}^H \boldsymbol{v}_m / (\boldsymbol{v}_{m-1}{}^H \boldsymbol{v}_{m-1})$, $\boldsymbol{v}_m = \boldsymbol{v}_m / \sqrt{(\boldsymbol{v}_m{}^H \boldsymbol{v}_m)}$ $(m=1, 2, \cdots)$。如果矩阵$\boldsymbol{A}$的特征值满足$|\lambda_1| < |\lambda_2| \leqslant |\lambda_3| \leqslant \cdots \leqslant |\lambda_n|$,那么$\theta_m$收敛到$\lambda_1$,$\boldsymbol{v}_m$以$|\lambda_2|/|\lambda_1|$的速度收敛到$\lambda_1$对应的特征向量。

如果目标特征值与μ的距离远小于其他特征值与μ的距离,且存在$(\boldsymbol{A}-\mu\boldsymbol{I})^{-1}$,那么在幂法中以$(\boldsymbol{A}-\mu\boldsymbol{I})^{-1}$代替$\boldsymbol{A}$,就得到反幂法。作为对幂法收敛性的改进,反幂法的迭代向量将快速收敛到目标特征值对应的特征向量。因此,反幂法常用于计算近似特征值的特征向量。进一步,在反幂法中通过Rayleigh 商更新位移量,就得到 Rayleigh 商迭代。Rayleigh 商迭代的渐近收敛速度是 2 阶,对于 Hermite 矩阵是 3 阶。

幂法保持了矩阵的稀疏结构,适用于大规模稀疏矩阵的并行计算。同时,幂法也为分析 Krylov 子空间迭代方法的收敛行为提供了基本模型,其意义已经超出了算法本身。

子空间方法 计算矩阵多个特征值及相应特征向量的一种迭代算法。幂法每次只限于计算一个特征向量,而幂法迭代向量的线性组合有助于找到多个特征向量。为此,对于给定矩阵$\boldsymbol{A}$和初始向量$\boldsymbol{v}$,引入 Krylov 子空间$\boldsymbol{K}_k(\boldsymbol{A},\boldsymbol{v}) \equiv \mathrm{span}\{\boldsymbol{v},\boldsymbol{A}\boldsymbol{v},\boldsymbol{A}^2\boldsymbol{v},\cdots,\boldsymbol{A}^{k-1}\boldsymbol{v}\}$。如果存在数$\theta$和非零向量$\boldsymbol{y} \in \boldsymbol{K}_k(\boldsymbol{A},\boldsymbol{v})$,对任一向量$\boldsymbol{w} \in \boldsymbol{K}_k(\boldsymbol{A},\boldsymbol{v})$成立$\boldsymbol{w}^H(\boldsymbol{A}\boldsymbol{y}-\theta\boldsymbol{y})=0$,那么$\theta$是一个 Ritz 值,$\boldsymbol{y}$是$\theta$对应的一个 Ritz 向量。假设$\boldsymbol{V}_k=(\boldsymbol{v}_1,\boldsymbol{v}_2,\cdots,\boldsymbol{v}_k)$是$\boldsymbol{K}_k(\boldsymbol{A},\boldsymbol{v})$的一个正交归一的基底,可通过求解一个$k$阶矩阵特征值问题$\boldsymbol{V}_k{}^H\boldsymbol{A}\boldsymbol{V}_k=\theta\boldsymbol{y}$来计算 Ritz 对$(\theta,\boldsymbol{y})$。此时,$(\theta,\boldsymbol{V}_k\boldsymbol{y})$给出了一个近似的特征对。这里的$k$阶矩阵特征值问题通常采用 QR 方法求解。

Arnoldi 分解是计算 Krylov 子空间$\boldsymbol{V}_k$的一种迭代算法。它从初始向量$\boldsymbol{v}$出发,通过$k-1$次迭代将n阶矩阵$\boldsymbol{A}$分解为如下形式:$\boldsymbol{A}\boldsymbol{V}_k=\boldsymbol{V}_k\boldsymbol{H}_k+f_k\boldsymbol{e}_k{}^H$,其中$\boldsymbol{V}_k$是一个$n\times k$的酉矩阵,满足$V_k{}^H f_k=0$;$\boldsymbol{H}_k$是一个$k$阶 Hessenberg 矩阵;$\boldsymbol{e}_k$是第$k$个分量为 1 的单位向量。如果$\boldsymbol{A}$是一个 Hermite 矩阵,那么$\boldsymbol{Hk}$是对称三对角实矩阵,此时得到的是一个$k$步 Lanczos 分解。$\boldsymbol{V}_k$的列向量称为 Arnoldi/Lanczos 向量,它们构成k维 Krylov 子空间$\boldsymbol{K}_k(\boldsymbol{A},\boldsymbol{v})$的一个正交基。$k$步 Arnoldi/Lanczos 分解的每个迭代步需要计算一个n阶矩阵向量乘积和两个不超过k阶的矩阵向量乘积,适于大规模稀疏矩阵的并行计算。

在实际应用中,我们往往只对矩阵$\boldsymbol{A}$的一部分特征值感兴趣。为了使 Arnoldi/Lanczos 分解产生的 Krylov 子空间包含更多感兴趣的特征向量信息,发展了重启动(restart)技术,即从子空间中提取一个所谓的重启动向量,然后从这个向量开始新的 Arnoldi/Lanczos 分解。显式重启动技术通过多项式预优构造重启动向量,从而削弱不感兴趣的特征向量组分。隐式重启动技术通过隐式位移 QR 迭代将一个较大的子空间压缩为一个较小的子空间。隐式位移 QR 迭代不仅有效过滤了不感兴趣的特征向量组分,同时具有良好的数值稳定性。

ARPACK(ARnoldi PACKage)是求解大规模稀疏矩阵特征值问题的并行计算软件包,其核心算法是隐式重启动的 Arnoldi/Lanczos 方法。

广义特征值问题数值解法 在某些情况下,如采用有限元方法离散特征值微分方程,导致一个广义的矩阵特征值问题$\boldsymbol{A}\boldsymbol{x}=\lambda\boldsymbol{B}\boldsymbol{x}$,其中$\boldsymbol{x}$是非零向量。它的数值解法是前面所述矩阵特征值问题的推广。当$\boldsymbol{B}$非奇异时,可化为$(\boldsymbol{B}^{-1}\boldsymbol{A})\boldsymbol{x}=\lambda\boldsymbol{x}$的矩阵特征值问题。当$\boldsymbol{A}$,$\boldsymbol{B}$均为 Hermite 矩阵且$\boldsymbol{B}$正定时,可化为$(\boldsymbol{L}^{-1}\boldsymbol{A}\boldsymbol{L}^{-H})(\boldsymbol{L}^H\boldsymbol{x})=\lambda(\boldsymbol{L}^H\boldsymbol{x})$的矩阵特征值问题,其中$\boldsymbol{B}=\boldsymbol{L}\boldsymbol{L}^H$,$\boldsymbol{L}$是非奇异下三角矩阵。当$\boldsymbol{A}$,$\boldsymbol{B}$均为大规模稀疏 Hermite 矩阵且$\boldsymbol{B}$正定时,可化为$(\boldsymbol{A}-\mu\boldsymbol{B})^{-1}\boldsymbol{B}\boldsymbol{x}=\theta\boldsymbol{x}$的矩阵特征值问题,其中$\mu$是位移量,$\lambda=1/(\theta+\mu)$。此外,还发展了一些直接求解广义特征值问题的数值方法,其中 QZ 方法是求解非对称矩阵广义特征值问题最有效的算法,它是矩阵特征值问题 QR 方法的推广。

参考文献

1. Golub G H, van Loan C F. Matrix computations. Maryland: The Johns Hopkins University Press, 1983

2. Saad Y. Numerical methods for large eigenvalue problems. New York: Halsted Press and John Wiley & Sons Inc., 1992

(李庆扬 周树荃 高兴誉)

juxing jisuanji

巨型计算机(supercomputer) 指在一定时期可用的、运算速度最快、性能最高的一种**计算机**。简称巨型机,又称超级计算机。它是计算机型谱中的最高档机型。巨型计算机主要用于解决**大型计算机**也难以解决的复杂问题。它是解决科技领域中某些巨大的挑战性问题的关键工具。

发展简史

现代巨型计算机起源于20世纪60年代末至70年代初。按照体系结构和技术水平的发展，巨型计算机已经历4代。

第一代巨型机是单指令流多数据流SIMD的**阵列处理机**，其典型代表是美国于1972年研制成功的ILLIAC-Ⅳ。它由64个处理机构成8×8阵列，这些处理机在一个控制部件的控制下同步并行工作。该机由分立元件组成，运算速度每秒近1亿次。第二代巨型机是具有流水线结构的向量机。20世纪70年代中后期美国推出的Cray-1是这一代巨型计算机的标志。该机由高速中、小规模集成电路组成，有12条不同功能的流水线运算部件，分为4组，可并行流水工作，峰速达160 MFLOPS（按两条流水线计）。该机使巨型计算机首次成为商品，并走向市场。第三代巨型机是多指令流多数据流MIMD的共享主存多处理机系统，以美国于20世纪80年代中期到90年代初推出的Cray X-MP、Cray Y-MP系列为代表，具有2～16个处理机，紧密耦合共享主存，可多任务并发以解决用户的大型问题。每个处理机内采用多流水线向量技术，运算速度达每秒几十亿到近二百亿次。第四代巨型机是大规模并行处理系统，起步于20世纪80年代初，由成千上万、甚至更多的处理器互连而成，靠高度并行来获得超高性能，从而实现每秒千亿次、万亿次、千万亿次以至更高的性能。典型系统有1993年美国的CM-5，含1024个处理器，峰值速度132 GFLOPS；1997年美国的ASCI Red，含7264个处理器，峰值速度1453 GFLOPS；2002年美国的BlueGene（蓝色基因），含32 768个处理器，峰值速度91 750GFLOPS；2011年日本的K Computer（“京”），含548 352个处理器核，峰值速度8 773 630 GFLOPS。大规模并行处理系统利用超大规模集成电路技术，硬件实现相对容易一些，性能价格比高，已经成为当前的技术主流。

组　成

巨型计算机通常由计算子系统、服务子系统、I/O存储子系统、监控诊断子系统、互连通信子系统以及基础架构子系统等硬件组成。计算子系统由大量的处理器或大量由若干处理器构成的计算节点组成，实现系统的主要计算处理功能；服务子系统实现登录、资源和作业管理以及外部连接等功能；I/O存储子系统实现海量数据的存储与输入输出；监控诊断子系统负责全系统的监控和维护诊断；互连通信子系统实现计算子系统、服务子系统、I/O存储子系统以及监控诊断子系统之间的互连通信；基础架构子系统实现系统的供电、散热和组装。

巨型计算机的软件通常包括操作系统、编译系统、并行程序开发与应用环境。

分　类

巨型计算机有多种不同的分类方法。根据系统中使用定制部件还是商用部件的程度，巨型计算机可以分为商用型、定制型和混合型三类。商用型巨型计算机，使用工业标准化的商用处理器和商用互连，通常称为机群（cluster）；定制型巨型计算机，使用专门设计的微处理器和专门设计的互连网络，具有比商用型巨型机高得多的本地存储器带宽和节点间带宽，典型系统有日本的Earth-Simulator（地球模拟器）；混合型巨型计算机使用商业微处理器和定制的高带宽互连网络，充分利用商业处理器的高性价比和定制互连网络（和定制处理器-存储器接口）的高访存与通信性能，典型系统有美国的ASC Red Storm。

根据所面向应用领域的计算复杂程度，巨型计算机可以分为能力型和容量型两类。能力型巨型机面向综合性能要求高的高端科学与工程计算领域，一般为定制型或混合型系统；容量型巨型机面向广泛的商业与工业应用，一般为商用型系统。

根据系统中使用处理器的多样性，可以分为同构型和异构型两类。同构型巨型机只使用一种中央处理器；异构型巨型机使用多种处理器，一般为“通用中央处理器+加速处理器”结构，典型系统有美国的Roadrunner、中国的“天河一号-A”。

巨型计算机技术

（1）体系结构　巨型机获得高处理速度的主要手段是并行技术。有两种并行，一是利用流水线原理，以时间重叠方式加工数据；二是多个处理部件以空间分布方式实现并行处理。在体系结构中要研究如何开发这两种并行，以构成一个合理、实用、高效的系统。多处理部件间的互连、同步和通信是技术难关。如何建立多处理部件互连拓扑与实际数学问题求解之间的映象，以充分发挥多处理部件的效率，是体系结构和并行算法相结合的重要研究课题。大容量存储器比运算器慢得多，如何使存储器和运算器的速度匹配是体系结构的一大难题。多级存储器是解决这一问题的传统途径。如何使输入输出能力和计算能力匹配是又一关键技术。

（2）工程实现　巨型机获得高处理速度的另一个主要手段是提高计算机的时钟频率，即提高处理

单元自身的处理速度和互连单元自身的通信速度。限制时钟频率提高的因素主要有两个：一是元器件的开关速度，二是元器件之间的连线长度。前者依赖于微电子技术的发展，后者依赖于器件集成度和高密度组装技术。巨型机的运算速度越高，其核心部分的物理尺寸应越小，这是高难度技术。巨型机庞大复杂，耗电量大，稳定均衡的供电是又一大难题。对于耗电量大而尺寸小的组装，散热是关键。

(3) 软件　巨型机的操作系统应是并行、分布式操作系统，具有批处理和一定的交互处理功能，它直接影响巨型机系统的稳定性和使用效率。并行编译器将高级语言编写的用户程序编译成能在巨型机的多个处理机或多条流水线上并行执行的程序，其关键是充分开发并行性，使程序得以高效地执行。软件工具和开发环境可以协助用户开发、优化、调试和分析其并行程序，它们是帮助用户用好巨型机的重要软件。并行分布式数据库、各种科学计算的算法库、图形库、高速网络软件、科学计算可视化软件等都是巨型机重要的软件。

(4) 并行算法　针对各种用户的复杂的实际问题，研究开发相应的并行算法是用好巨型计算机的又一个重要方面。算法和体系结构匹配可以提高巨型机的实际使用性能。

(5) 性能测评　使用评价指标(metrics)和基准测试(benchmarks)技术对系统性能和面向应用的实际使用效能进行衡量，可以对已有系统进行评价，也可对设计系统的指标进行预测。

发展趋势

科学技术的发展和社会信息化对巨型计算机的计算能力不断提出新的要求，每秒千万亿次运算的巨型计算机已经成为现实，百亿亿次巨型机也将在2020年前问世。微电子技术已经能在1 cm^2的硅片上集成几亿个晶体管，商品化的64位微处理器的速度已达每秒几千亿次浮点计算。基于大规模并行处理技术，以大量处理器为基础构造巨型机仍将是重要的技术途径。从使能技术的角度，使用速度更快的处理器、容量更大带宽更高的存储器、带宽更高延迟更低的互连与交换技术，以不断提高系统性能。从应用效能的角度，由于系统规模的不断增大以及各种技术发展的不平衡，必须通过创新的体系结构和软件技术，以实现系统的高效率、高可靠、低成本和易使用(参见**高效能计算机**)。

参考文献

1. Hwang K. Advanced computer architecture: parallelism, scalability, programmability. New York: McGraw-Hill Inc., 1993

2. http://www.top500.org

（杨学军）

jufa moshi shibie fangfa

句法模式识别方法(syntactic pattern recognition method)　一种基于结构关系，以形式语言为工具的模式描述和分析方法。

用计算机进行模式识别(参见**模式识别**)可以归纳为两种处理方法：统计(或称决策理论)方法和句法(或称结构)方法。统计方法是从模式中抽取一组能代表模式特性的测量值(称为特征)，并用划分特征空间的办法来对模式进行分类。对于一些模式识别的问题，描述模式的结构信息具有重要意义。对模式的识别不仅要求能够把输入的模式指定到一个特定的类别，即把它分类，而且要对那些不至于把输入模式进行误分类的特征加以描述。在这类问题中，所研究的模式通常是十分复杂的，需要的特征也很多。因此，用一些比较简单的子模式组成多级结构来描述一个复杂的模式，就成为人们研究的方向。为了表示多级结构信息，可以将一个模式用一些比较简单的子模式来描述，而每一个简单的子模式再用更加简单的子模式来描述，正如英语中的短语和句子由单词连接而成，单词又由字符连接而成一样。实质上，模式识别的句法方法从一个方面展示了模式的结构与语言的句法两者间的相似性质。为发挥句法方法的优越性选定的最简单的子模式，即“模式基元”，应比模式本身更容易识别。通过一组模式基元及其组合运算来描述模式结构的“语言”称为“模式描述语言”。由基元构成模式所遵循的那些规则，可以用生成模式描述语言的“文法”来确定。识别的过程是通过对描述给定模式的“句子”进行句法分析来确定该句子在句法(或文法)上是否正确。模式的描述，既类似于语言的构成，又并不局限于像语言那样，字符之间仅是左右的连接，而是可以扩展到高维的连接与描述，从而把对模式的研究转换为对模式基元构成的串、树或图的研究。

句法方法的构思始于20世纪70年代初。第一本系统地对句法方法加以论述的著作是美国Purdue大学傅京孙(K. S. Fu)教授于1974年在美国出版的《模式识别的句法方法》。该书奠定了句法模式识别方法的基础。以前句法方法并没有得到广泛的应用，主要问题在于用形式语言描述模式，即使对简单的模式，也必须采用“**上下文有关文法**”；而识别模

式则要对这种类型的文法进行句法分析,而上下文有关文法的分析是十分复杂的,因而句法方法的应用受到限制。经过研究改进,着眼于把统计方法和句法方法两者有机地结合起来,以属性文法为基础,引入基元的属性,基元间、基元与子模式间的联接属性,规定一些"语义规则",利用语言描述中语法和语义之间存在的一种"折衷关系",通过增加语义描述的复杂程度而使语法描述变得简单,从而克服了以往句法方法在应用中所遇到的困难,并且利用人工神经网络形成了一种"语义、句法方法"。由于吸收了统计方法与句法方法的优点,拓宽了句法方法的应用范围。这种方法已经成功地用于手写汉字识别这类结构信息起着重要作用的领域。

参考文献

1. Fu K S. Syntactic methods in pattern recognition. Academic Press, Inc. , 1974

2. Tai J W. Semantic syntax-directed translation for pictorial pattern recognition. Proc ICPR. Münich, 1982

3. Tal J W. A connectionist syntactic semantic approach. Proc Int'l Conf on Information Sciences. Seoal, 1993

(戴汝为)

juli tuxiang fenxi

距离图像分析(range image analysis) 对距离图像(参见距离图像获取)进行处理、抽取特征、建模、识别的过程。距离图像是一类直接表达三维空间中物体表面点之间几何位置关系的数字图像,其像素值表达的是传感器与景物点间的距离。

利用距离图像获取方法(参见**距离图像获取**)得到的距离图像有以下特点:①噪声影响严重,且难以用简单的统计方法建模。②数据不完整。由于大多数点云数据存在采样不均匀、数据漏洞(有些位置没有数据)较多等问题,所生成的距离图像不完整,或经插值处理后的数据变形严重。③场景中物体形状与结构复杂,致使距离图像严重依赖于观测视点。

距离图像的分析包括以下内容:

(1)预处理与特征抽取 距离图像的预处理与特征抽取主要有以下过程:去噪、分割、平滑区域的曲面拟合、区域间相互关系的描述。曲率是表示距离图像中曲面形状变化的主要特征之一,而经常使用的曲面形状描述有扩展高斯图像与曲面旋转图像等。

(2)三维形状匹配与基于模型的物体识别 与二维灰度或彩色图像相比,距离图像更好地反映了三维景物的本质特征,有较强的视点与光照不变性。通常使用的场景分析策略是:对于给定的三维物体模型,将其与图像描述匹配,以确认模型在图像中是否存在,若存在,则计算出其位姿参数。

(3)三维建模 获取距离图像的另一目的是建立三维物体或场景的几何模型。建模过程主要包括:①网格生成与简化、配准、整合。②将从不同视点采集的距离图像转换到同一个空间坐标系中,以得到模型的完整表示。③将转换到同一坐标系的网格进行整合,以减小不同视点数据间的微妙差异,消除数据接缝对网格结构造成的影响。

参考文献

1. Cyganek B, Siebert J P. An introduction to 3D computer vision techniques and algorithms. Wiley, 2009

2. Ma Y, Soatto S, Kosecká J, Sastry S S. An introduction to 3D vision: from images to geometric models. New York, Springer, 2004

(查红彬)

juli tuxiang huoqu

距离图像获取(range image acquisition) 利用摄像机成像原理,通过估计单幅或多幅图像的深度,进而完成图像场景的距离图像的计算过程。

距离图像是一类直接表达三维空间中物体表面点之间几何位置关系的数字图像,其像素值表达的是传感器与景物点间的距离。距离感知是人类日常生活中不可或缺的认知机理之一。同样,距离图像也是智能机器系统实现对环境感知与交互的重要信息来源。目前,距离图像在机器人导航与作业、目标跟踪、智能人机交互、游戏影像制作等方面得到广泛的应用。

大多数情况下我们所直接获取的数据是空间中三维采样点的集合,称作三维点云。这些数据具有以特定空间坐标系为基准的三维坐标。但由于存在噪声大、密度分布不均匀、邻接关系不明确等问题,这些数据必须经过一定的后处理才能转换成通常意义下的距离图像。

三维数据获取方法可分为被动式与主动式两大类。前者是不对场景的空间结构与光照环境带来任何的人工干预,主要是基于图像的成像原理与摄像机间的几何关系来计算景物点三维位置;而后者则需要投射一定的扫描光或结构光图案,或者改变光照条件,再利用这些附加约束来求解场景的深度值。

被动式数据获取方法包括:①立体视觉 通过

处理和融合两个不同视点所得到的立体图像对，计算物体表面的深度。如果我们能够在立体图像对中同时观测到任一景物点的投影，则利用三角测量原理，可以算出景物点相对于视觉系统的空间位置。②多视点重建　增加视点的数量，利用多视点图像的信息冗余，提高场景深度数据的密度与完整性。此外，图像或图像序列中的许多视觉特征都可在被动式三维重建中得到应用，如图像中直线群的透视结构、纹理的密度变化、随着焦距调整的图像清晰度变化（散焦测距法）等。

通常使用的主动式方法有：①基于立体匹配的深度扫描　向景物表面投射便于区分的光源，使立体图像对中的对应点得到标记。为了保证深度重建过程的效率，可利用不同形式的投射光图案或扫描策略，以求在更短的时间内得到更多的重建数据。②基于飞行时间的深度扫描　当传感器发射激光或超声波脉冲并能回收其反射信号时，依据信号传播的时间就可简单地计算出传感器与物体表面间的距离。一般，当需重建的景物大小超过一定范围后，立体视觉的视差太小，基于飞行时间的扫描就成了不可替代的数据获取技术。

参考文献

1. Cyganek B, Siebert J P. An introduction to 3D computer vision techniques and algorithms, Wiley, 2009

2. Ma Y, Soatto S, Kosecká J, Sastry S S. An introduction to 3D vision: From images to geometric models. New York, Springer, 2004

（查红彬）

juece shu

决策树（decision tree）　基于分治策略的归纳学习所得结果的表示形式（参见**机器学习**），又称判定树。它从根结点到某个叶结点划分示例集。叶结点则为某个示例的分类类别。决策树的每个内结点表示测试示例的某个属性，该结点的每一后继分支对应该属性的一个可能取值。从数据中产生决策树的过程称为决策树学习算法。

决策树学习过程即是从树的根结点开始，通过测试结点对应的属性值对示例集逐级划分，直到一个结点仅含有同一类的示例为止。最早的决策树学习算法是概念学习系统 CLS，最著名的是 Quinlan 的 ID_3 和 C4.5，以及 CART 和 ASSISTANT。决策树学习已被成功应用于医疗诊断、信贷风险评估等领域。

分类属性的选择决定了算法的效率和所生成决策树的繁简程度，属性选择是决策树归纳算法的关键：

CLS 系统在选择属性时没有特定的启发式策略。

ID_3 是 CLS 的改进，使用“信息熵”来选择划分属性。C4.5 在 ID_3 的基础上有了进一步提高，一方面通过使用“信息增益率”来改进属性划分准则，另一方面可以对连续属性进行学习，因此得到广泛应用。

CART 由分类树和回归树两部分构成，分别用于示例类别是离散变量和数值变量的问题。

ASSISTANT 主要特点是使用类似算法 AQ_{15} 的方法进行近似匹配，其中 AQ_{15} 是一种以产生式规则表示的归纳学习算法。

决策树的优点是分类和测试速度快，特别适用于大数据库的学习问题。其局限是：①决策树的**知识表示**没有规则形式好，特别是较大的分类树意义费解；②比较两棵决策树是否等价是子图匹配问题，属 NP 完全问题；③同一类的叶结点分布分散，影响近似匹配的精度。

寻找一棵最优决策树，主要应解决以下 3 个最优化问题：①生成最少数目的叶结点；②生成的每个叶结点的深度最小；③生成的决策树叶结点最少且每个叶结点的深度最小。

目前，围绕决策树的**机器学习**主要研究以下问题：①避免决策树学习中的过度拟合；②新的属性选择度量标准；③连续属性离散化处理；④属性值缺失训练示例的处理；⑤与示例属性关联的代价考虑；⑥决策树的修剪与分支合并操作。

参考文献

1. Mitchell T M. Machine learning. New York: McGraw-Hill, 1997

2. Duda R O, Hart P E, Stork D G. Pattern classification. 2nd ed. New York: John Wiley & Sons. 2001

3. 洪家荣. 归纳学习——算法 理论 应用. 北京：科学出版社，1997

（郭茂祖）

juece zhichi xitong

决策支持系统（decision support system, DSS）　以人机交互方式为决策人员提供辅助决策信息的**应用软件系统**。DSS 是**管理信息系统**的延伸和发展，目的是帮助决策者提高决策的科学性和正确性。

1970 年，美国麻省理工学院的 S. S. Morton 等人根据计算机对决策的支持作用，提出决策支持系统的概念。1980 年 Sprague 提出了决策支持系统三部件结构（对话部件、数据部件、模型部件），明确了 DSS 的基本组成。DSS 第一次国际学术讨论会于 1981 年举行。20 世纪 80 年代末 90 年代初，DSS 开

始与**专家系统**相结合形成智能决策支持系统（IDSS）。这一阶段，DSS利用**专家系统**以知识推理形式解决定性分析问题，以模型计算为核心解决定量分析问题，实现了定性和定量分析的有机结合，使得解决问题的能力和范围得到快速发展。90年代中期以来，随着**数据仓库**、联机分析处理、**数据挖掘**、业务智能等技术的出现，DSS逐渐发展成了一种更高级形式的综合DSS（synthetic decision support system，SDSS）。综合DSS不仅用模型和知识辅助决策，而且能从大量数据中发现知识，并获取辅助决策信息，因而可以实现更有效的辅助决策。

决策按其性质可分为3类：①结构化决策　指对某一决策过程的环境及规则，能以确定的数学模型描述，能用适当的算法产生决策方案，并能从多种方案中选择最优解的决策；②非结构化决策　指求解的问题含有许多不确定因素，且决策过程复杂，不可能用确定的数学模型描述其决策过程，更无所谓最优解的决策；③半结构化决策　指介于以上二者之间的决策，这类决策尽管可以建立适当的算法产生决策方案，但决策数据是不完全或不精确的，因而只能从可选方案中得到较优解或满意解的决策。非结构化和半结构化决策通常服务于中、高管理层决策。

结构化决策一般分为4个步骤：①发现问题并形成决策目标，包括建立由决策树法、表格法、矩阵法或各类规则（如关联、聚类、转移）表示的决策模型，拟定决策方案和确定效果度量；②用概率定量地描述每个方案所产生的各种结局的可能性；③决策者对各种结局进行定量评价，一般用效用值来定量表示，效用值是决策者根据个人才能、经验、风格以及所处环境条件等因素而对各种结局的价值所作的定量估计；④综合分析各方面信息，以最后决定方案的取舍，有时还要对方案作灵敏度分析，研究原始数据发生变化时对最优解的影响，决定对方案有较大影响的参量范围。决策往往不能一次完成，特别是对于非结构化或半结构化决策，通常需要一个迭代过程和频繁的人机交互过程。一方面决策者需要通过DSS来辅助确定目标、拟订方案、分析评价和模拟验证，另一方面决策者需要为DSS提供各种不同方案的参量，并根据经验对DSS给出的各种辅助决策信息自行进行分析判断和方案选择，直至及时做出正确有效的决策。

DSS的基本结构如图1所示，主要由4个部分

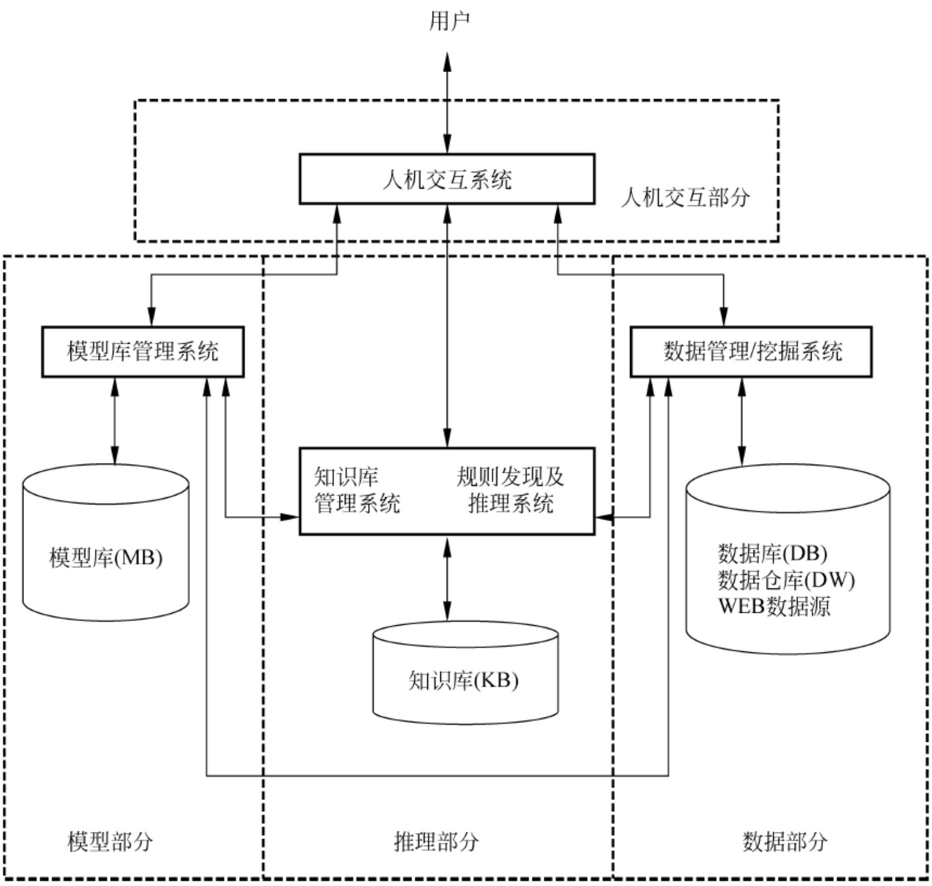


图1　决策支持系统基本结构

组成,即数据部分、模型部分、推理部分和人机交互部分。数据部分包括**数据库**、**数据仓库**和Web数据源及其数据管理/挖掘系统。模型部分包括模型库及其管理系统。推理部分由**知识库**、知识库管理系统和规则发现及**推理机**组成。人机交互部分用以检验和接受用户输入的信息或命令,向用户展现DSS给出的各种辅助决策信息以及需要用户回答的各种请求。

当前,DSS主要用于企业预测和分析、计划和销售、研究与开发等职能部门,也有用于社会科学、宏观经济调控、市场和投资效益分析等方面,在军事、工程和区域规划等领域同样有广泛的应用。

随着软件技术的快速发展和计算机硬件平台的不断更新换代,DSS融入了网络、中间件、多媒体和面向对象等技术,通过数据资源、模型资源、知识资源等决策资源的共享,其功能从简单的决策支持已发展到在网络环境下支持处理高度复杂和智能的半结构化和非结构化的决策问题,其决策支持能力从主要支持个人决策发展到支持群体决策活动,并形成了具有灵活支持能力和提供并发共享服务的新体系,出现了分布式DSS(DDSS)、群体DSS(GDSS)、空间DSS(SDSS)、自适应DSS(ADSS)、决策支持中心(DSC)、战略DSS和集成化智能交互型DSS(I^3DSS)等应用系统。

参考文献

1. 埃弗雷姆·特班,杰伊 E. 阿伦森,梁定澎. 决策支持系统与智能系统(原书第7版). 杨东涛,钱峰,译. 北京:机械工业出版社,2009

2. 李东,梁定澎. 决策支持系统与商务智能. 北京:中国人民大学出版社,2010

(孙志挥)

junshi zhihui xinxi xitong

军事指挥信息系统(military command information system) 以计算机系统为基础,运用先进的信息技术进行获取、传输、处理和利用各种军事信息,并进行辅助决策,对部队实施指挥控制及战场管理的信息系统。通常所说的C3I、C4I和C4ISR等系统都是军事指挥系统的一些具体工程实现。

军事指挥信息系统的出现始于20世纪50年代。1953年,美国开始研制以计算机为中心的自动化指挥系统,并于1958年率先建成了世界上第一个自动指挥控制系统,首次将地面警戒雷达、通信设备、电子计算机和显示器连接起来,实现了目标航迹绘制和其他数据显示的自动化。20世纪60年代,随着远程武器的发展,特别是各种战略导弹和战略轰炸机的大量装备部队,指挥决策与作战行动执行单位之间可能彼此相隔数千千米甚至更远,单一的指挥控制系统已无法胜任现代战争的指挥与控制任务,无法实时地进行大量信息的传输。为了改变这种状况,美军又逐步建立了一批战略、战术指挥自动化系统,如全球军事指挥控制系统。从20世纪70年代初到80年代末,随着计算机技术的飞速发展,计算机渗透到指挥自动化系统乃至武器系统的各个节点、末梢以及一切与信息有关的环节,实现了信息采集、分析与方案制定、辅助决策等高层次信息处理活动的自动化。在这一时期内,美军研制了一系列军事指挥信息系统,如空中预警系统,空中指挥所系统,地下指挥所系统,机动型战术空军的军事指挥信息系统,以及三军的联合战术信息分发系统。自20世纪90年代至今,军事指挥信息系统进入了综合一体化发展的新阶段。

军事指挥信息系统的组成从技术设备的角度考虑,可分为传感器系统、通信系统、数据处理系统、显示控制系统和技术保证系统。从系统功能的角度考虑,可分为指挥控制系统、预警探测系统、情报侦察系统、通信系统、电子战系统和战场支援保障系统。从战场信息流程的角度考虑,可分为信息获取与感知系统、信息传输与分发系统、信息分析与处理系统、信息开发与利用系统、信息安全与对抗和信息存储与显示系统。

信息获取与感知系统主要包括预警探测系统和情报侦察系统,解决的问题是要看得到、看得清,这是指挥决策的基础。预警探测系统是军事指挥信息系统最重要的实时信息源,它直接影响到观察、判断、决策、行动和整个军事行动的全过程。情报侦察系统主要功能是平时和战时搜集各种情报信息,为各级指挥员提供决策依据,为作战部队提供作战信息,这是取得战争胜利的重要保障。信息获取与感知系统将向全天候、全信息影像、多信源综合、微型化、智能化和强生存能力方向发展,形成无处不在的战场感知。

信息传输与分发系统是战场信息互通和共享的纽带,主要包括通信系统和数据链系统,解决的问题是连得通、传得快。通信系统由通信装备、设施和通信人员组成,用以组织通信联络、保障军队指挥的系统,其作用是把指挥系统诸要素联结成一个有机的整体。数据链系统是利用无线信道机,在各种飞机、水面舰艇、陆基武器及不同战术平台之间,实时、自

动、保密地传输、分发和交换格式化战术数据的特殊数据通信系统,其作用是实现指挥系统与武器系统的无缝隙连接和战场信息共享。信息传输与分发系统将向一体化、宽带化、多功能、抗干扰方向发展。

信息分析与处理系统主要是解决算得精、判断准的问题,其核心是计算机及其软件,其关键是能够近乎实际地向各级指战员提供战场空间内敌我双方的态势信息,通过数据融合、情报融合与态势汇合等技术形成通用作战图像,使指挥员掌握战场态势,拟制作战方案。

信息开发与利用系统是军事指挥信息系统的核心,主要包括指挥控制系统,解决的是结合紧、用得好的问题。指挥控制系统是指军队的各级指挥所(中心)系统,指挥员通过数据处理、显示控制和辅助决策等对部队实施作战指挥和管理。

信息安全与对抗系统是其他各个系统的基础,主要作用是实现人机交互,保证指挥人员能对整个作战进程全面地、不间断地、有效地指挥和控制。

未来的军事指挥信息系统将朝着数字化、网络化、自动化、智能化和一体化的方向发展,通过一体化建设,所有部队处在一个共同的指挥控制系统之下,拥有信息优势,使各军兵种能像一支单独的部队发挥作用,确保诸军兵种联合作战的协同指挥。

(李德毅)

K

katong donghua

卡通动画(cartoon animation) 用计算机制作卡通风格动画的技术和过程。其分类有计算机辅助制作二维卡通动画与计算机三维卡通动画。

卡通原是英文 cartoon 的音译,指报纸或期刊上讲笑话或以幽默方式进行讽刺批评的漫画。在美术中则指一种粗糙的草图,其大小与最终完成的结果图一样。卡通动画是画出一系列表现动作的画面,由于人眼的视觉暂留作用,若以每秒 24 帧或更高的速度播放这些序列画面就可以形成连续运动的感觉。在卡通动画中角色的造型和动态一般比较简单并且夸张,其中以迪士尼风格动画为代表。在我国动画片中还包括剪纸片、水墨动画以及折纸片,在视觉效果上具有独特的风格。

传统的卡通动画制作是相当费时费力的,一部长篇动画需要许多人员参与:导演、制片人、动画设计人员和动画辅助制作人员。为了提高效率和降低成本,动画制作要遵循一定的程序,具体包含 16 个步骤:剧本创作,故事板编写,声音记录,历程分解,造型设计,动画设计,中间画面绘制,Leica 卷片,试验线稿,整理,描线与着色,背景,检查,拍摄,样片和配音复制。计算机辅助制作二维卡通动画就是用计算机完成其中的某些步骤,如输入画面、着色、数字编辑、背景绘制、声音记录、试验线稿、曝光表和中间画面生成等。中间画面生成的原理是给出一个动画角色在第 M 幅和第 N 幅画面上的姿势,在它们中间的 $M+1$ 到 $N-1$ 画面由计算机计算生成出来,最常用的是线性或样条插值方法。

借助计算机图形技术还诞生一种三维卡通动画,在这里物体仍然采用光照模型进行真实绘制,但动画角色的造型和动作却比较简单和夸张,其中以《玩具总动员》为代表。近年来出现的**非真实感图形绘制**技术也为生成不同风格的二维和三维卡通动画提供了基础。

参考文献

1. Rick Parent. Computer Animation: Algorithms and Techniques. 2nd ed. San Fransisco, CA: Morgan Kaufmann, 2007

2. Isaac V Kerlow. The Art of 3D Computer Animation and Effects. 4th ed. Wiley Publishing, 2009

(于金辉)

kaifa guocheng

开发过程(development process) 软件开发人员所负责的一系列活动,其目的是依据合同成功地开发并交付软件。当要开发新的系统,或对已有的系统进行版本升级以及对已有系统进行有开发活动的移植时,都要涉及开发过程。

开发过程包含的活动有:需求分析、设计、编码、集成、测试、安装以及验收支持等,也可包含与系统有关的活动。开发过程同时还贯穿软件过程中其他过程的实施;开发人员通过**管理过程**管理开发过程;通过**剪裁过程**剪裁开发过程的活动;通过改进过程参与开发过程的管理(参见**软件过程**);通过基础设施过程在开发过程中建立基础设施(参见**软件过程**);通过**支持过程**的一些过程实施开发过程中的文档编制、联合评审、审计等。

以下是开发过程所涉及的活动,其中活动(1)是必需的,且是应当首先完成的,活动(2)至活动(13)是可选的,开发人员应当首先完成活动(1)的任务,然后根据活动(1)所产生的结果选择活动(2)至活动(13),完成开发过程。

(1) 本过程的实施准备 目的是为开发过程准备最基本的约定。其主要任务有:①依据合同及软件或系统的特点选择活动(2)至活动(13),所选择的活动可重复或相关联,亦可循环交替;②制定本过程计划,计划中至少包含所需的内部标准、方法、工具、行为、责任和所使用的程序设计语言等;③指定各种文档的编制方式,安排其他各**支持过程**的实施方法。

(2) 系统需求分析 目的是根据合同分析系统的具体应用意图,完成系统需求。系统需求规约的内容有:①对系统的功能和性能的要求,包括安全、保密性;②人机界面、操作和维护需求;③设计方面的限制等。检查该系统需求规约是否达到以下要求:能否跟踪**获取过程**得到的系统需求并保持一致;

是否具备可测试性;是否具备系统结构设计的条件;操作与维护是否可行等。

(3) 系统结构设计　目的是建立一个高层的系统体系结构。该体系结构将系统需求按硬件、软件、人工操作这三个要素分为硬件配置项、软件配置项和人工操作项。检查该结构是否达到以下要求:能否与系统需求保持一致;所使用的方法是否符合标准;所确定的软件配置项需求是否可行;操作和维护是否可行等。

(4) 软件需求分析　目的是确定软件需求及质量特性需求,并完成需求分析规约。软件需求分析规约的内容包括: ①功能和性能需求; ②外界与软件(即软件配置项)的接口; ③合格需求; ④安全需求,包括与操作、维护、环境等对人员的伤害有关的说明; ⑤保密需求; ⑥人机工程和人机界面需求; ⑦数据定义和数据库需求; ⑧现场安装及验收需求; ⑨用户文档; ⑩用户操作和运作需求; ⑪用户维护需求等。检查该软件需求是否达到以下要求:能否跟踪系统需求和系统结构;是否从外部与系统需求保持一致;软件需求内部是否具备一致性;测试覆盖程度是否达到要求;是否具备可测试性;操作和维护的可行性如何等。

(5) 软件体系结构设计　目的是将软件需求分析规约转化为一个体系结构。其主要任务有: ①定义并描述该结构的顶层部分; ②为软件外部接口、数据库、软件各部件之间的接口作顶层设计; ③设计用户手册的初级版本; ④定义初步测试需求并制定软件集成计划等。检查该结构是否达到以下要求;能否与软件需求保持一致;结构内各部件是否具备一致性;所用设计方法与标准是否一致;是否具备进一步作详细设计的条件;操作与维护是否可行等。

(6) 软件详细设计　目的是详细设计软件体系结构中的每个部件。主要任务有: ①尽可能将每个部件细划为可进行编码、编译及测试的软件单元; ②详细设计软件的外部接口、软件各部件之间的接口、各单元之间的接口; ③详细设计数据库; ④充实用户手册; ⑤定义单元测试需求并制定单元测试计划; ⑥充实集成测试需求并制定集成测试计划等。检查该详细设计是否达到以下要求:能否与软件需求和体系结构保持一致;各部件以及各单元之间是否具备一致性;所使用的设计方法是否符合标准;是否可测试;是否具备易操作性和易维护性等。

(7) 软件编码和测试　主要任务有: ①根据详细设计结果进行编码,提供测试数据,完成单元测试; ②充实软件集成测试需求和集成计划; ③充实用户手册等。检查本活动是否达到以下要求:能否与软件需求和软件设计保持一致;在软件部件内部,各单元需求之间是否具备一致性;单元的测试覆盖程度是否达到要求;编码方法是否符合标准;是否具备软件集成及测试的条件;是否具备易操作性和易维护性等。

(8) 软件集成　目的是制定计划,将上述活动得到的各软件单元和软件部件和从获取过程得到的软件集成为所需软件。主要任务有: ①制定计划,计划中至少包括测试需求、步骤、数据、责任和时间进度表等内容; ②按计划进行软件的集成; ③充实用户手册; ④针对合格需求制定合格测试集及步骤等。检查本活动以及集成好的软件是否达到以下要求:能否与系统需求保持一致;内部是否具备一致性;测试所用的方法是否符合标准;是否符合预期结果;是否具备软件合格测试的条件;是否具备易操作性和易维护性等。

(9) 软件合格测试　目的是完成软件的合格测试。依据合格需求进行合格测试,并充实用户手册。检查本活动是否达到以下要求:软件需求的测试覆盖度是否达到要求;是否符合预期结果;系统集成和测试是否可行;是否具备易操作性和易维护性等。可能时,应支持审计工作,审计后应准备一套完整的软件,交付给活动(10)至活动(13)。

(10) 系统集成　目的是将交付来的软件集成到系统中。主要任务有: ①将软件同有关硬件、人工操作项以及其他必要的系统集成为一个系统; ②为系统合格测试进行必要的准备,包括开发测试集和测试步骤。检查集成后的系统是否达到以下要求:系统需求的测试覆盖程度是否达到要求;所用测试方法是否符合标准;是否符合预期结果;是否具备系统合格测试的条件;是否具备易操作性和易维护性等。

(11) 系统合格测试　主要任务是依据合格需求和合格测试计划进行系统合格测试。检查本活动是否达到以下要求:系统需求的测试覆盖是否达到要求;是否符合预期结果;是否具备易操作性和易维护性等。可能时,应支持审计工作,并在审计结束后准备一套完整的软件以便进行软件安装或软件验收支持。

(12) 软件安装　目的是在目标环境中安装软件。主要任务有: ①制定安装计划,包括与软件安

装有关的信息和资源；②实施软件安装，注意保证该软件和数据库能按合同的规定初始化、运行和终止。

(13) 软件验收支持　目的是支持获取者对软件的验收评审和测试。主要任务有：①支持验收评审和测试；②按合同交付文档及代码；③依据合同向获取者提供培训和支持。

参考文献

1. IEEE Standard for Developing Software Life Cycle Processes — IEEE Std 1074—1991. 1991

2. ISO /IEC 12207:1995 Information Technology— Software — Part 1: Software Life-Cycle Process. 1995

（宿为民）

kaifang xitong

开放系统(open system)　遵循公开定义的接口和协议，便于与其他系统互连的**计算机系统**。负责开发易移植操作系统接口(POSIX)标准的美国电气和电子工程师协会 IEEE P1003 工作组把开放系统定义为：按照开放的接口、服务和支持的规范而实现的系统。开放系统能使：①应用软件基本上不作修改就可实现在不同系统间的移植；②本地或远程系统的各应用间实现互操作；③各应用间可方便地实现交互作用。

开放系统所遵循的标准和规范必须是公开的，因为这是共同遵守、维护和适时更新的前提。IEEE P1003.0《POSIX 开放系统环境指南》把开放规范定义为：同国际标准一致的规范，并且以开放的、多数同意的过程进行维护，以便能随时适应新技术的发展。

开放系统环境的程序界面、人机界面、系统管理工具、互操作服务、通信服务和安全性等方面都是按所选用的公开标准(国际标准、工业标准或事实标准)实现的，从而使得在这种环境中的各计算机平台间能确保应用软件的可移植性、可剪裁性和可互操作性，在这种环境中的用户计算机资源能得到最大限度的保护。因此，标准是开放系统的依据；可移植性、可剪裁性和可互操作性是开放系统的特征；对用户利益的保护是开放系统的目的。

按照开放系统的目标所设计的计算机体系结构称为**开放体系结构**，用开放系统的概念和开放体系结构的计算机所构成的应用系统称为**开放应用系统** OAS。

开放系统的形成与发展

回顾计算机科学技术和应用的发展历史，在网络计算尚未得到大规模的推广应用前，计算机系统主要由单机或由同类型计算机构成。有影响的计算机公司都拥有自己独有的操作系统，那时的产品虽然也宣称具有可互联、可共享等性能，但都以自我为中心。

随着计算机技术和通信技术的发展，以往那种独特的、排他的产品已无法适应信息互通、信息共享等需求。为了适应大规模推广计算机的应用和计算机网络化的需求，尤其是要把多台不同类型的、不同地点的计算机集成为一个系统，需要尽量提高软件的可复用性和可移植性，因此，提出了一系列的问题：如何能方便、有效地把多台不同类型的计算机连接在一起？如何确保集成后的系统能协调、高效地工作？如何能在今后的系统中，保证已有的硬、软件资源可继续使用和共享？

总之，大家认识到，今后计算机系统的发展策略必须从过去的以我为主的唯我型改变为遵循标准、开放兼容、资源共享的开放型。因此，可移植性、互操作性、易集成性等就成为开放系统所追求的基本目标了。为实现这一目标，首先是建立一个可以确保不同机器和系统间的互操作性、软件的可移植性和由低向高的资源可继承性(规模可伸缩性)的公开标准。有了这样一个公开标准后，用户就可能以最佳的性能价格比，从多个供应商那里优选计算机的硬、软件资源，构成与其他系统兼容和资源共享的系统。

为了能最终实现一个理想化的开放系统，用户从应用的角度出发，希望有一个标准化的计算机应用环境，在这个标准化了的计算应用环境中，用户可方便地实现不同应用系统间的互操作、应用系统的移植和剪裁以及计算资源的高度共享。从图 1 可见，一个理想的计算机应用环境，其中的每一部分都是遵照公开标准实现的。

世界上有许多组织机构能对标准的形成产生影响，如由政府组织的标准化机构、由特定应用领域的用户组织的标准化机构以及由工业界的生产厂家组织的标准化机构。ISO(国际标准化组织)、ANSI(美国国家标准学会)和我国的国家技术监督局等均为政府组织的标准化机构。在国际上，也存在一些由生产厂家组成的标准化机构，它们为了使自己的产品能与其他产品互连和兼容，根据自己的利益和能

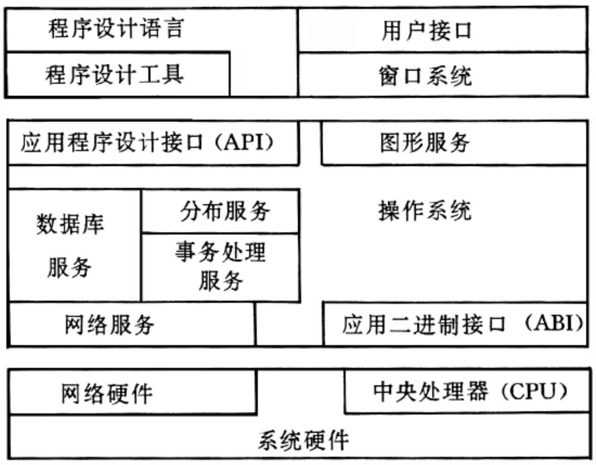


图 1　理想的计算机应用环境

力制定某些标准。其中较有影响的有:UI,X-Open,OSF, COS 等。

开放系统的标准

国际上已公布或正在讨论和制定中的有关开放系统的标准大体上分为 3 类:

(1) 基础标准　负责制定功能标准的特别工作组 ISO/IEC JTC1/SGFS 在 TR 1000 技术报告中把基础标准定义为:为了在系统间交换信息而制定的一组单层或多层的规程和格式。

最有代表性和权威性的开放系统的基础标准是由国际标准化组织(ISO)所提出的**开放系统互连基准(参考)模型**。该模型定义了一种抽象结构,共由 7 层组成,即:物理层、数据链路层、网络层、运输层、会话层、表示层和应用层。该模型的特点是:①定义了一种能将异构系统互连的分层结构;②提供了控制互连系统间交互作用规则的标准框架;③定义了在不同计算机的同层之间实现通信的协议规程。

OSI 及相关标准是由 ISO 和国际电工委员会(IEC)两大国际标准组织的联合技术委员会 JTC1 负责制定的,由 JTC1 下属的有关分技术委员会(SC)和工作组(WG)具体负责开发这些标准,具体分工如下:

ISO/IEC JTC1/SC 6	系统间通信和信息交换
WG 1	数据链路层
WG 2	网络层
WG 3	物理层
WG 4	运输层
ISO/IEC JTC1/SC 18	文本和办公系统
WG 1	用户需求和 SC 18 管理
WG 3	文档体系结构
WG 4	文本交换系统
WG 5	内容体系结构
WG 8	文本描述和处理语言
WG 9	用户-系统接口和符号
ISO/IEC JTC1/SC 21	开放系统互连的信息检索、传送和管理
WG 1	OSI 体系结构
WG 3	数据库
WG 4	OSI 系统管理
WG 5	特定应用服务
WG 6	OSI 会话、表示和公共应用服务
WG 7	开放式分布处理(ODP)
ISO/IEC JTC1/SC 27	用于信息技术应用的公共安全技术
WG 1	密钥算法及应用
WG 2	公钥加密系统及使用模型
WG 3	通信体系结构中使用加密技术
ISO/IEC JTC1/SGFS	功能标准

OSI 及相关标准已超过 200 项,详细说明了协议规范和每一位的定义。

以下要介绍的 POSIX 属于基础标准。

(2) 功能标准　当用户在具体构建某个实际的开放系统时,应在基础标准的规范下,根据用户的实际需求和已具有的网络资源和能力,具体制定功能标准,或称为功能轮廓或规范文件。

OSI 在制定基础标准时,由于已预先考虑了各种不同的应用需求以及各式各样的网络结构,所以提供了多种功能选项。功能标准是基础标准的具体化,不能与基础标准相矛盾,只是根据具体情况对基础标准所允许的选项加以特定的选择。每个国家、企业或公司都可根据自己的具体情况对基础标准所提供的可选项进行设定,从而制定出各自的功能标准。

许多国家的政府部门均作出决策,要求其公共部门必须采购并使用 OSI 产品。美国政府机构的 OSI 功能轮廓 GOSIP 已作为联邦政府信息处理标准 FIPS 146,于 1988 年正式发布。

英国的中央计算机和电信局(CCTA)定义了名为 MUSIC 的开放系统应用结构框架,其中的 M,U,S,I,C 分别表示管理、用户接口、系统和应用接口、

信息和数据服务、通信服务。

以下要介绍的 X-Open，OSF 等属于功能标准。

(3) 应用标准　当用户基于开放系统的概念，在具有开放体系结构的计算机系统上开发各自的开放应用系统时，应按照应用标准，通过 3 种不同的抽象级别对该应用系统进行描述，即需求描述、过程描述和代码描述。

开放系统标准的制定涉及计算机系统的各个组成部分，如中央处理器、操作系统、网络通信协议、数据库、各种服务功能和用户接口等。从用户需求出发，首先应解决的是用户接口和网络软、硬件的标准化问题，使用户可方便地、无阻碍地使用网络内的全部资源。从系统的技术实现出发，操作系统的标准化无疑是最核心的问题。

由于国际市场上的绝大多数计算机系统都使用 UNIX 操作系统，因此，UNIX 也就成了实现开放系统的基本操作系统。在 1984 年，市场上曾存在着多种版本的 UNIX。最有影响的是以下 3 种：由AT& T 公司提供的 UNIX System Ⅴ，由 Berkeley Software Distribution 提供的 BSD UNIX 和由 Microsoft 公司提供的 Xenix。如何统一这些已流行的 UNIX 系统成为实现开放系统的一个关键问题。

基于 UNIX 的开放系统规范

(1) POSIX　美国 UNIX 用户团体为统一 UNIX 标准曾编制了一个 UNIX 的统一规范，称为 POSIX (可移植的操作系统接口)。1988 年，经 IEEE 和 ISO 讨论后认定它为国际标准，并在原基础上进一步明确：POSIX 是指由 IEEE 1003.0 ~ 1003.9 工作组开发的一套标准，目的是保证应用系统的可移植性。其中：

IEEE 1003.0 工作组定义应用可移植性环境的元素、标准及其集成；

IEEE 1003.1 工作组定义 POSIX.1 系统接口；

IEEE 1003.2 工作组说明了 POSIX.2 shell 和工具；

IEEE 1003.3 工作组定义 POSIX.3 测试和测证；

IEEE 1003.4 工作组定义 POSIX.4 实时扩充；

IEEE 1003.8 工作组制定 POSIX.8 网络文件系统(NFS)，提供透明的文件访问。

(2) OSF　开放软件基金会(OSF)是由 135 家计算机制造商和研究机构组成的非盈利组织，其目的是为开放软件环境制定一套应用环境规范(AES)，并对按照这些规范开发的源程序发放许可证。

(3) UI 和 X-Open　UI 是由 AT& T 公司和它的 UNIX 特许单位组成的专门对 UNIX 操作系统进行定义和控制的机构。1993 年决定由 X-Open 继续原 UI 的标准化工作。X-Open 成立于 1984 年，发起者以欧洲的计算机厂家为主，目的是促进 UNIX 操作系统标准化。它是一个非盈利性组织，任务是在 ANSI，IEEE 和 ISO 等标准化组织所正式公布的标准中选定自己的开放系统规范。

X-Open 和 OSF 都不是制定基础标准的机构，它们只是从 ANSI，IEEE 和 ISO 等标准化组织正式公布的标准中，选择和决定开放系统需要的规范。表 1 扼要地给出了 X-Open 和 OSF 的开放系统规范。

表 1　X-Open 和 OSF 的开放系统规范

	开放软件基金会 OSF(应用环境规范 AES)	X-Open(X-Open 可移植性指南 XPG 3)
操作系统 POSIX	ISO 9945-1 IEEE 1003.1 ISO 9945-2 IEEE 1003.2	ISO 9945-1 IEEE 1003.1 ISO 9945-2 IEEE 1003.2
用户接口	X-Window 系统 OSF/Motif	X-Window 系统
图形	GKS　ISO 7943 GKS-3D　ISO 8805 PHIGS　ISO 9592	未作规定

续表

	开放软件基金会 OSF(应用环境规范 AES)	X-Open(X-Open 可移植性指南 XPG 3)
网络	TCP/IP OSI XTI (X-Open 传输接口)	XTI (X-Open 传输接口)
数据管理	SQL ISO 9074	SQL ISO 9074 ISAM 终端接口源码交换
语言	FORTRAN ISO 1539 C ANSI X.3 159 PASCAL ISO 7185 COBOL ISO 1989 Ada ANSI/MIL 1815A LISP COMMON LISP BASIC	FORTRAN ISO 1539 C ANSI X.3 159 PASCAL ISO 7185 COBOL ISO 1989 Ada ANSI/MIL 1815A

(4) 在 POSIX 基础上,OSF 与 X-Open 的结合　为了真正建立一个基于 UNIX 操作系统的开放应用环境,UI(以及 X-Open)和 OSF 都一直在促进建立符合 POSIX 规范的结合。具体的作法见图 2,在共同遵循 POSIX 的基础上,OSF 的操作系统(如 OSF/11.1)和 UI 的操作系统(如 SVR 4.2)相互采用对方的基本技术。系统的操作系统是基于微核心软件 Mach 3.0 之上的;在 Mach 之上建立大量面向各种主流机型(如使用 OSF,BSD,ULTRIX,SVR4,VMS,MVS,DOS,OS/2,Windows 等的机型)的服务软件;由 OSF 提供中间件,包括分布式计算环境(OSF/MOTIF/DCE)和分布式管理环境(OSF/MOTIF/DME)。用户的应用软件是在中间件之上构造的。通过中间件确保应用软件的可移植性。X-Open 决定把它所开发的 XPG(X-Open 可移植性指南)同分布式计算环境 DCE 相一致,从而促进了 OSF 与 X-Open 的结合。但离用户所希望的、如图 2 所示那样的一个开放的应用环境还有很大的距离。

理想开放系统的实现是一个长期追求的目标,是真正实现信息共享的必要前提。制定并遵循开放系统的规范和标准是实现开放系统的依据。由于实现开放系统的技术问题尚未全部解决,各个集团和公司都在争夺制定各种标准的主导权和维护各自的利益,各用户在淘汰(或改造)旧系统、转向开放系统时还存在习惯、资金和人力等因素,因此,开放系统的实现将是一个逐步完善的过程。

参考文献

1. Nutt G J. Open Systems. Prentice Hall,1992

2. Gray P A. Open Systems—A Business Strategy for the 1990s. McGraw-Hill,1991

(汪成为)

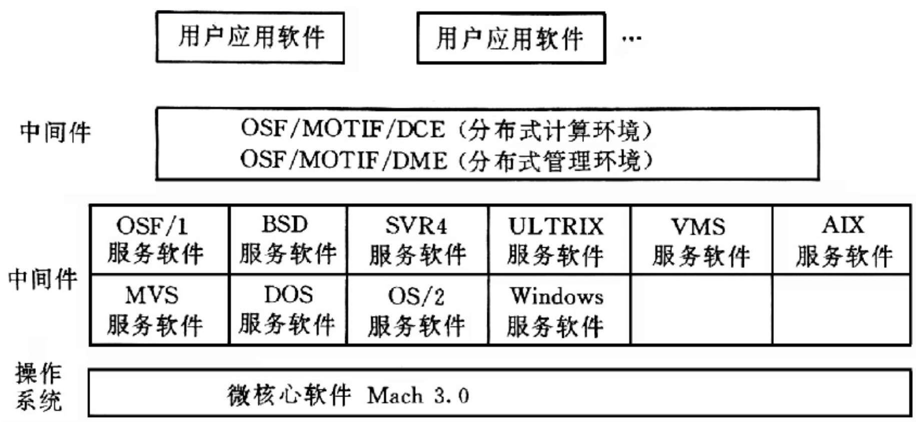


图 2　结合 OSF 与 X-Open 的具体做法

kaifang xitong hulian jizhun (cankao) moxing

开放系统互连基准(参考)模型(open system interconnection reference model) 由国际标准化组织(ISO)提出和定义的**网络体系结构**,简称**OSI 基准(参考)模型**(ISO/OSI 7498),如图 1 所示。

OSI 基准(参考)模型于 1977 年被 ISO 的 TC(技术委员会)提出以来,ISO 又分别为各层制定了协议标准,从而形成了完整的 OSI 网络体系结构。在 OSI 网络体系结构中,低层协议为相邻的高层协议提供相应的服务,高层协议作为低层协议的用户存在。OSI 网络体系结构的七层协议是:

(1) 物理层 物理层在通信信道上传输原始比特。物理层协议被设计来控制传输介质,从而提供传输介质以及和这些介质相连接的机械的、电气的接口。这些接口和传输介质必须保证发送和接收信号的同一性。

(2) 数据链路层 数据链路层协议加强物理层原始比特的传输功能。数据链路层把数据分装在不同的数据帧中发送,并处理接收方回送的确认帧。数据链路层通过在帧的前头和末尾附加上特殊的二进制编码来产生和识别帧界。数据链路层协议必须保证传输和接收的数据帧的正确性以及发送速度与接收速度的匹配。数据链路层协议还完成流量控制和出错处理工作。

(3) 网络层 网络层关系到对通信子网的运行控制。网络层负责选择从发送端传输一个数据包到达接收端的通信路径,起中继作用。网络层还负责通信子网中的分组,拥塞控制和记账等。路径选择方法通常有两种,即固定路径选择表方式和根据网络负载情况动态选择的方式。网络层协议有面向连接的协议和无连接的协议两种。它们向高层提供面向连接的网络层服务和无连接的网络层协议。

(4) 传输层 传输层是 OSI 协议层次结构中最核心的一层。它把实际使用的物理网络与高层应用分开,提供发送端和接收端之间的高可靠性、低成本数据传输。传输协议为会话层提供面向连接的传输服务和无连接的传输服务。为了提供性能可靠和价格合理的数据传输,传输层协议必须完成寻找接收端用户地址,提供面向连接服务时的建立连接、撤销连接以及流量控制和**多路复用**等工作。

(5) 会话层 会话层协议属于 OSI 基准(参考)模型的高层。它使用传输协议提供的可靠的端到端通信服务,并增加一些用户需要的附加功能和建立不同机器上的用户之间的会话关系。会话层协议为表示层提供同步服务,使得低层协议在发生了某种错误之后,会话层协议能返回到一个已知状态。会话层还为表示层提供活动管理功能。活动是一个由用户确定的具有逻辑意义的信息单位。会话层协议的另一个重要功能是数据交换。

(6) 表示层 表示层协议完成被传输数据的表示和解释工作,包括数据转换、数据加密、数据压缩等。表示层协议的主要功能有:①为用户提供执行会话层服务原语的手段;②提供描述复杂数据结构方法;③管理当前所需的数据结构集;④完成数据的内部格式与外部格式间的转换。

(7) 应用层 应用层包含大量人们普遍需要的协议。例如虚终端协议、文件传输系统、远程用户登录和电子数据交换以及**电子邮件**等。

在 OSI 网络体系结构中,高层协议用户通过服务访问点(SAP)和低层协议发生作用。服务访问点是服务使用者和服务提供者之间的界面。*N* 层服务访问点记作 NSAP。同一层中的不同服务访问点用标识符来区别,服务访问点标识符称为地址。相邻高层协议和低层协议的相互作用通过服务原语实现。

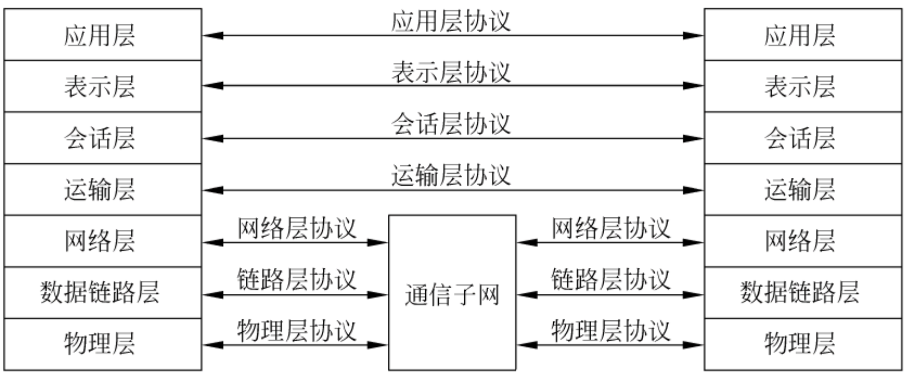


图 1 OSI 网络体系结构

服务原语是对服务提供者和服务使用者相互作用的最基本的行为的描述,它描述服务提供者和服务使用者一次最基本的交互作用的名称以及各参数的意义。

使用 OSI 网络体系结构时,除了物理层之外,网络中数据的实际传输方向是垂直的。用户发送数据时,首先由发送进程把数据交给应用层,应用层在数据的前面加上该层有关控制和识别信息,再把它们交给表示层。这一过程一直重复到物理层,并由传输介质把数据传送到接收端,在接收进程所在的计算机中,信息向上传递时,各层的有关控制和识别信息被逐层剥去,最后,数据被送到接收进程。数据传输时数据变化过程如图 2 所示。

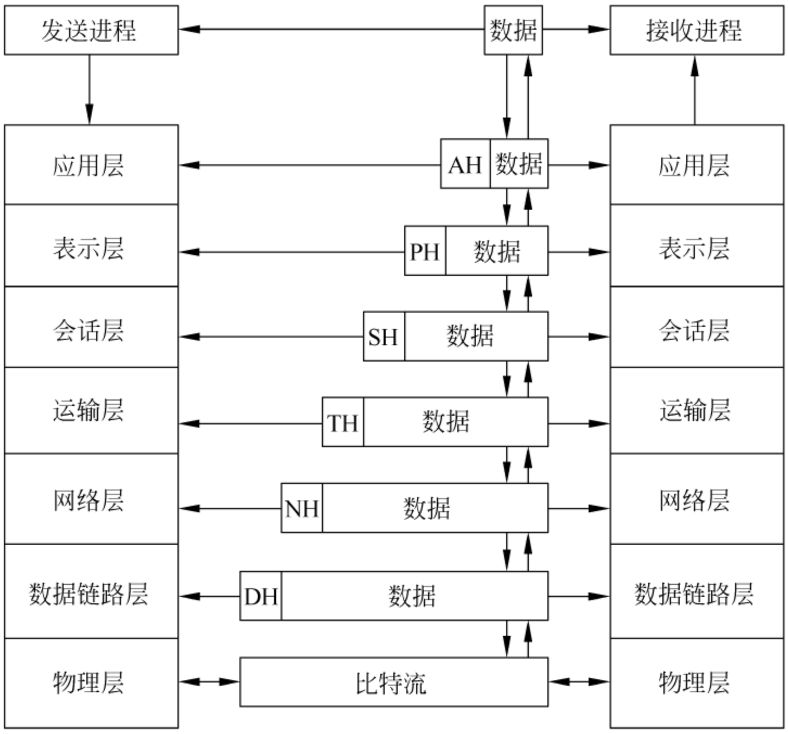


图 2 OSI 网络体系结构中数据变化过程

参考文献

1. ISO/OSI 7498. The OSI Reference Model. 1978

2. 张尧学,等. 计算机网络与 Internet 教程. 2 版. 北京: 清华大学出版社, 2006

3. Tanenbaum AS, Wetherall DJ. 计算机网络. 5 版. 严伟,潘爱民,译. 北京: 清华大学出版社, 2012

(张尧学 相士俊)

kaiyuan ruanjian

开源软件(open source software) 一种源代码可以自由获取和传播的计算机软件,其许可人在开源许可证的规定之下允许被许可人对软件进行使用、修改和传播。开源软件也称为自由/开源软件(free/open source software)。相比专有软件,开源软件发展出全新的软件分享规则、开发方法和商业模式,并在软件技术和商业领域取得巨大成就,因此业界常常将开源软件的发展过程称为开源运动和开源革命。目前,业界公认的开源软件的定义由开源软件促进协会(OSI)给出(现为 1.9 版)。该定义认为符合以下 10 项条款的软件可称为开源软件:①自由再发行;②必须包含或方便取得源代码;③许可证必须允许更改或派生程序;④保护作者源代码的完整性;⑤无个人或团体的歧视;⑥无领域歧视;⑦依据许可证发行;⑧许可证不能特指某个产品;⑨许可证不能约束其他软件;⑩许可证必须独立于技术。

历史沿革

开源软件概念的形成和发展大致经历了自由软件运动兴起、开源运动兴起和开源软件发展三个阶段。第一阶段是自由软件运动兴起阶段,主要时间跨度为 20 世纪 60 年代至 90 年代中期。自由软件运动的目标是尽可能确保软件用户对软件的使用、修改和传播的自由,反对以保护知识产权和商业利益为目标的专有软件对这些自由的限制。自由软件

运动起源于20世纪60～70年代美国计算机科学实验室中的黑客文化，以1985年Richard Stallman发起的自由软件基金会（FSF）为标志。其中，1984年Richard Stallman启动GNU项目，拉开了自由软件运动的序幕，1989年FSF发布通用公共许可证第一版GPLv1，为自由软件运动制定了行为规则。1991年Linux的发布以及GNU/Linux的广泛应用使自由软件运动在软件领域产生了实质性影响。

第二阶段是开源运动兴起阶段，主要发生在20世纪90年代中后期，以1998年开源（Open Source）术语的提出和OSI对开源软件的明确定义为标志。由于自由软件对知识产权的排斥及其许可证（如GPLv1）关于自由的传染性使大多数商业公司持观望态度，人们开始探索能够平衡开放理念和商业利益的新模式。1998年Netscape公司发起的一次战略研讨会上，Christine Peterson首次使用开源一词来回避自由软件在商业领域的负面效应。随后Eric Raymond提出使用开源软件来统称所有符合OSI定义的软件，开源运动进入一个崭新的阶段，众多商业公司加入进来，各种开源软件许可证不断涌现，为开源软件的商业化发展开辟了道路。

第三阶段是开源软件发展阶段，主要从20世纪90年代末开始至今，越来越多的个人和商业公司通过开源软件建立了自身的开发方法和商业模式，开源软件在技术创新和商业运营等方面扮演着越来越重要的角色。从1999年至今，开源软件取得了商业上令人瞩目的成就，如1999年Red Hat等开源软件公司成功上市、IBM启动对Linux的支持服务，2001年IBM直接将Eclipse项目贡献给开源社区，2005年Google收购开源软件公司Android并迅速实现了移动业务的迅猛拓展。此外，以1999年SourceForge上线为标志，基于互联网的各种开源软件开发和社交平台开始快速发展，吸引了全球数百万的开发者和用户，支撑了数十万的开源项目的开发，为开源软件的全球化协同开发和运营提供了支撑环境。

开源许可证

开源许可证是一种特殊的软件许可证，采用契约和授权方式指导和规范许可人和被许可人在处理开源软件作品时的权利、义务和责任，也是解决开源软件面临的法律和商业问题的核心机制。目前通过OSI认证的开源许可证已达数十种，其中应用最为普遍的典型许可证主要有GPL、BSD、LGPL和Apache许可证等。这些许可证满足OSI开源软件定义中关于许可证的共性要求，但在处理开放性、商业用途、知识产权、兼容性等问题方面存在差别。例如，GPL要求根据GPL发布的软件衍生出来的作品也必须遵循GPL；Apache许可证的专利许可证的授予条款能够有效避免Apache及其用户卷入知识产权诉讼；BSD则允许被许可人根据自己的需要（包括以商业软件的方式）再发布和再许可等，只要求被许可人附上BSD原文以及所有开发者的版权资料。

开源组织

众多的开源组织是支持开源软件事业发展的主要推动者，包括开源基金会和开源社区等。其中，自由软件基金会（FSF）和开源软件促进会（OSI）属于推动自由/开源软件整体性发展的非盈利性组织，从自由和开源两个出发点在发展方向、规则制定、商业战略等方面发挥了启蒙和重大推动作用；Linux Kernel社区、Apache软件基金会（ASF）、Eclipse社区、Mozilla社区等开源组织则立足于具体的开源软件项目，为一系列开源项目提供规则制定、法律支持和财政帮助，极大地推动了Linux和Apache等开源软件生态环境的形成和商业模式的繁荣。

开发方法与基础设施

开源软件的开发方法是一种以开放、对等、共享和全球协作为核心的新型群体协作方法。同传统软件开发方法相比，该方法在降低开发成本、提升软件质量、减少培训和维护费用等方面具有先天优势。基于该方法，分布在世界各个角落的软件开发者和用户在开源组织中少数核心人员的协调下，利用业余时间成功开发出诸如操作系统Linux、服务器软件Apache、开发环境Eclipse和浏览器Mozilla Firefox等众多商业级软件作品。Eric Raymond在其“The Cathedral and the Bazaar”一书中将开源软件的开发方法比作“集市”模式，深刻而生动地揭示了开源活动中自由松散表象下隐藏的惊人软件工程新理论，并通过自身实践验证了开源方法的基本原则。

目前支持开源软件开发和发展的基础设施主要包括三类。①社区网站：开源组织为了实现旗下开源软件的快速开发、共享和发展，常常建立专门的社区网站为开发者和用户提供开发基础设施，包括项目主页、版本管理、沟通工具、缺陷跟踪系统、邮件列表、项目论坛、Wiki等。目前代表性的社区网站包括Linux Kernel、Eclipse和Mozilla等。②托管网站：托管网站为开源组织和自由开发者提供统一的基础设施，托管网站为大量开源项目提供上述基础设施服务。与社区网站不同，托管网站一般并不拥有开

源项目。目前代表性的托管站点包括 SourceForge、Google Code 和 GitHub 等。③目录网站:目录网站对众多的开源软件的信息要素进行收集和分类,并将社会化标签、用户评价等社交网站的机制引入开源目录网站,极大地提高了海量开源软件资源的检索、分析和选择效率。目前代表性的目录网站包括 ohloh 和 Freecode 等。

应用范围与商业模式

目前开源软件在商业、政府、教育甚至国防等领域获得广泛应用并取得巨大成功,出现了众多基于开源软件的应用系统,其应用范围正在不断扩大。很多国家和政府甚至通过立法强制采用开源软件以实现节约开支、增强软件系统的安全性和可靠性、减少对软件厂商的依赖、发展自主软件研发能力、加强知识产权保护、改造软件产业等众多目标。

随着开源软件事业的不断发展,越来越多的公司和组织采用了新的基于开源软件的商业模式来获得商业目标。目前比较流行的开源软件的商业模式包括:增值产品、技术支持、咨询服务、广告收益、软硬件结合、双重授权、社区模式等,例如,Red Hat 是基于 Linux 的增值产品厂商和咨询服务商,IBM 为众多开源软件提供技术支持和方案咨询,Mozilla 的 Firefox 成功利用其客户端的广告优势实现盈利,Google 利用开源项目 Android 实现软硬件结合的盈利模式,MySQL 则为用户提供 GPL 和商业许可证的双重授权模式(无论用户选择哪一种许可证,MySQL 都会直接或间接的获益)。最后,社区模式是最能反映开源软件理想的商业模式,如 Linux、Eclipse 和 Apache 等开源组织就成功地利用社区模式发展旗下的开源项目,从而为用户提供更好的软件产品和技术支持,并通过毋庸置疑的高品质软件作品获得社会的信任和资助,领导业界的发展趋势。

发展趋势

从本质上讲,开源软件是人类历史上一次基于群体智慧的大规模协作的重大实践。开源运动的先行者通过对早期自由软件活动的深刻认识,创造性地解决了开源软件面临的法律和商业方面的障碍,逐步建立起基于群体智慧的软件开发方法和生态环境,将分布在全球的群体智慧汇集到开源软件作品中,把用户对高品质软件的需求、企业商业战略、抑制技术垄断、产业良性循环等诸多目标有效地集成到开源活动中,最终实现了对软件产业的根本性改造。这种认识世界和改造世界的循环发展的过程还将继续在开源世界生机勃勃地开展下去。

参考文献

1. Eric Steven Raymond, The Cathedral and the Bazaar, Version 3.0, 2000

2. Open Sources: Voices from the Open Source Revolution, 1st Edition, January 1999

3. 孙玉芳,武延军,等. 自由软件与开放源码运动文集,云南大学出版社,2004 年 11 月

4. 蔡俊杰,吕晶,等. 开源软件之道,电子工业出版社,2010 年 4 月

(王怀民 尹刚)

kang elie huanjing jisuanji

抗恶劣环境计算机(computer in severe environment, CSE) 在特种应用背景下,在恶劣环境中能正确运行程序,并能输出正确结果的计算机系统。所谓特种应用常常是指军事应用、实时控制、野外探测、野外科学考察、航空、航海、航天等专用应用系统。显而易见,这类应用系统所处的环境是非常恶劣的。这些恶劣的环境条件如果按其性质来分类,可分为物理性质的、化学性质的、生物性质的、地理性质的。如果按环境特点来分类,常常可分为如下几种类型:

(1) 恶劣的自然环境,例如高温、低温、潮湿、干燥、风雨、雷电、低气压、盐雾、沙尘、霉菌、昆虫、动物等。

(2) 恶劣的场环境,例如静电、强电磁场、电磁干扰和攻击、火花、粒子辐射、太阳辐射等。

(3) 恶劣的应力环境,例如振动、倾跌、翻倒、碰撞、跌落、加速度、冲击等。

(4) 恶劣的人为环境,恶劣的人为环境可能使系统莫名其妙地失效;也可能使系统丧失安全能力。人海中的计算机面临着来自两个方面的挑战,其一是人为的非恶意性差错,例如错误地操作、输入错误的信息等;其二是人为的恶性攻击,例如信息入侵、病毒攻击、利用系统的信息泄露来攻击系统等。

恶劣环境下计算机所面临的问题

计算机是一个很脆弱的系统。任何一个故障都有可能导致系统失效。对于计算机的故障而言,可分为内部故障和外部故障。对于内部故障,主要通过**容错计算**技术予以屏蔽和在线维修,以保证系统的正确运行。外部故障,则主要是由上述恶劣环境所引起的故障。由此可见,抗恶劣环境计算机(CSE)的相关工作,将主要集中在如何应对恶劣环境从外部作用到计算机系统使其产生故障的问题。

恶劣环境所引发的计算机故障,可以分为三种

类型：永久性故障、间歇性故障、偶然性故障。经验表明，针对永久性故障的检测与诊断，具有一定的有利条件，因为它们较容易被“捉住。”而对于间歇性和偶然性故障来讲，难于检测，尤其难于诊断，因为它们要么没有规律；要么“一闪而过。”基于大量统计表明，永久性故障的发生率仅占不到 10%，而后两类故障却要占 90%，甚至更多。因此，“抗恶劣环境”大量而艰难的工作，必将主要集中在如何应对计算机在恶劣环境下的间歇性和偶然性故障问题上。自然的，这两类故障解决得好，永久性故障也就迎刃而解了。

主要技术措施

为了适应恶劣的环境，计算机将采用必要的技术措施。在实际应用中，就一台具体计算机而言，并不要求同时能应对上述的所有恶劣条件，而是面对不同的应用背景和需求，根据具体的环境特点，有选择地采取相应措施。通常采用的技术措施主要有两种：容错技术和避错技术。

容错技术是在恶劣环境已经造成了计算机内部故障的情况下，通过冗余设计，利用检错和纠错技术，消除故障影响，保持系统正确运行。

避错技术是在设计、加工、应用的全过程中，力争屏蔽恶劣环境的影响，使其不会引起计算机的故障。

计算机抗恶劣环境的主要技术措施是这两种技术的结合。但是，从体积、功耗、加工、成本等各方面综合考虑，用得最多的是避错技术，也就是针对恶劣环境采用屏蔽、隔离等技术。这类系统技术可归纳为如下几个方面：

(1) 加固技术　在现有的商用计算机基础上，按国家标准，通过加固实现抗恶劣环境计算机系统。这是节省成本、缩短研制周期的一种办法。所采用的加固措施是一种综合性的。在元器件、电路、子系统、系统集成等各结构层采用相应的技术措施，例如：解决环境温度过高、过低以及计算机内部散热问题的热设计技术；解决应力问题的抗振、抗冲击设计技术；解决电磁干扰、粒子辐射等问题的电磁兼容设计技术及防护技术；解决信息泄露问题的防泄露技术，包括滤波、抑制泄露源技术；抗腐蚀、昆虫、鼠害问题的防护、涂层技术；隔离及密封技术等。

(2) 信息安全技术　包含密码技术、防火墙技术、网络安全等系统安全技术，解决人为的恶意攻击问题。这已经成为备受关注的专门研究领域。

(3) 健壮性设计　主要解决人为错误及人为的恶性攻击。

(4) 系统的高可维修性设计　针对可维修系统，提高系统可测试性及诊断精度，提高在线维修能力，保障系统在极短时间内完成修复。

(5) 军用计算机系统技术　按军标的要求，满足武器系统及作战需求，实现军用抗恶劣环境计算机系统。这也是一种综合性的系统技术，几乎包含了上述各项技术。

主要环境技术指标

对于 CSE 而言，除了性能指标必须满足要求之外，还要适应恶劣的环境条件。按照国家标准的要求，针对某一项环境条件的技术指标，都要对应一定的严酷等级以及相应的背景。这里选择如下几项常遇到的指标：

(1) 温度　最高严酷等级的高温存储温度为 70 ℃(48 h)，高温工作温度为 40 ℃，低温存储温度为 −55 ℃(24 h)，低温工作温度为 −20 ℃。

(2) 气压　严酷的存储低气压为 57 kPa，工作低气压为 57 kPa，时间要求超过 1 h。

(3) 淋雨　2 h 以上，淋雨在 10 cm/h 以上。

(4) 湿度　相对湿度为 95% 左右。

(5) 自由跌落　根据不同的严酷等级、系统重量、包装条件、跌落次数、跌落表面材料，在跌落高度分别为 25 mm、50 mm、100 mm、250 mm、500 mm、1000 mm 中选择相应高度作为自由跌落的指标。有的情况下，要求系统能够同时适应多项环境条件。例如，低温和低气压、高温和高湿等。在相应的国家标准中，可以详细地查看到有关技术指标。

发展趋势

(1) CSE 的网络化将进一步加深　大量的 CSE 应用系统将以网络形式(无线及有线)发挥作用，或作为网络的前端或作为中继。

(2) 多学科多领域技术交叉的进一步深化，CSE 系统技术水平会有大幅提高　各个领域所取得的新成果，在材料、设备、机理等各方面的相互结合必将推动 CSE 技术的发展。

(3) 系统在进一步小型化、轻型化、低成本的同时提高性能。

(4) 新型的系统结构将推动 CSE 新的发展　开放式系统设计、多核芯片水平的提高、新型体系结构都将会大力地支持新型的 CSE。

(5) 系统综合属性的进一步提高　不仅可靠性，而且综合可用性、可维修性、失效安全性、完整性、私密性都会有很大地提高。国际上把这种综合

属性统称为 dependability。

(6) 现有的 CSE 应用系统还将有相当长的应用周期　CSE 不是一种普通的计算机系统，它的应用是以需求为牵引，并不追求时尚；它的加工复杂、周期长、应用大系统需要长期稳定。这样，专用芯片的国产化将成为关键。

参考文献

刘恩德，张淑萍. 军用抗恶劣环境计算机发展现状及趋势. 见：中国计算机学会学术工作委员会，主编. 中国计算机科学技术发展报告 2005，北京：清华大学出版社，2015：26-41

（杨孝宗）

kexue shujuku

科学数据库（scientific database）　针对科学数据的特点和科学领域的需求，支持科学数据的存储、组织、查询和分析的数据管理系统。科学数据是指在科学实验、科学观察、科学分析和计算过程中产生的数据。

科学数据库的设计需要充分考虑科学数据的特点。科学数据不仅仅包含数值本身，而且包括丰富的元数据和语义信息，如科学数值的测量单位、科学数据产生的环境和实验设置、科学数据的数据精度和准确性等。同时，由于科学数据来源于不同的领域，科学数据带有很强的异构性。

科学数据库需要具备海量数据的处理能力和数据压缩技术。随着科学仪器的发展，科学数据采集的频率越来越高，科学数据库需要应对海量数据的挑战。同时，科学数据中元数据信息丰富，不同类型的科学数据其元数据彼此不同，科学数据实例中可能存在大量的空数值或重复数值。数据压缩技术能够有效减少这些数值导致的存储空间的浪费。

科学数据库应方便灵活地支持新的数据类型和其上的数据操作。由于科学数据结构灵活，采用关系模型或者树状模型不再合适。目前研究较多集中于科学数据的多维数据模型。同时，科学数据库中数据操作的设计需要考虑科学数据的应用场景，应支持科学数据的采样、估值和新的数据连接等操作。

数据分析和智能推理技术是科学数据库的重要特色。科学数据库需要将不同来源的数据有机集成，基于不同领域背景知识的模型，通过数据分析和科学推理，自动发现隐藏的、有价值的知识。科学数据库应允许用户表达和建模领域知识，引入数据分析和推理的方法。

数据质量管理是科学数据库设计中需要考虑的一个重要因素。由于仪器故障、不同的采集精度、环境的变化、不同数据采样的频率和人员失误等多种因素，很难保证采集到的科学数据的准确性。科学数据库系统的设计需要考虑这一因素，在实现科学数据加载、查询执行和数据分析推理过程中，验证科学数据的有效性、提高查询和分析结果的可用性。

参考文献

Liu L, Ozsu M T. Encyclopedia of database systems. Springer, 2009

（杨冬青　高军）

kebiancheng kongzhiqi

可编程控制器（programmable controller, PC）　一种以**微处理器**为核心，综合了计算机技术、自动控制技术和通信技术的现代工业控制装置。可编程控制器被称为现代工业自动化的三大支柱（PLC、CAD/CAM、机器人）之一，通常也称为**可编程逻辑控制器**（programmable logical controller, PLC）。

一台（套）可编程控制器通常由中央处理单元（CPU）、存储器、输入输出接口、扩展接口、通信接口、电源、编程器（编程软件）等几个部分组成，如图 1 所示。其中，输入输出接口又分为开关量输入/输出、模拟量输入/输出等类型，开关量输入/输出接口的作用是对“0”/“1”逻辑信号进行电平转换，模拟量输入接口的作用是把电流（mADC）、电压（VDC）、毫伏电势（mVDC）、欧姆（Ω）等连续变化的检测信号转换成 CPU 可以处理的数字信号，模拟量输出接口的作用是把数字信号转化成标准电流或电压信号，用来驱动调节阀、变频器等现场设备。扩展接口用于输入输出接口的扩展，或者多机架的扩展。通信接口用于和其他设备或控制系统实现通信连接，常见的通信接口包括：RS-232 接口、RS-422/RS-485

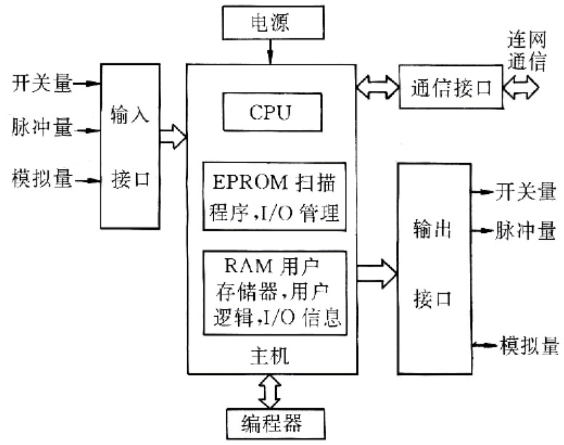


图 1　可编程控制器构成图

接口、以太网接口以及各种现场总线接口等。编程器(编程软件)主要用来给可编程控制器编制和修改程序,也可以用来监视可编程控制器的工作状态。

可编程控制器主要有两种结构形式,一类是一体化结构,即中央处理单元、存储器、输入输出接口、通信接口、电源等都集成在一个机壳内;另一类是模块化结构,即中央处理单元、输入输出接口、通信接口、电源等在结构上相互独立。一体化结构的可编程控制器多为微、小型可编程控制器,主要用于单机自动化,必要时也可以进行少量的 I/O 扩展;模块化结构的可编程控制器多为中、大型可编程控制器,可根据具体的应用要求,选择不同型号、不同数量的模块,以积木方式构成更大规模、更强功能的控制系统。

可编程控制器之所以取得快速发展和广泛应用,除了工业领域的客观需要外,主要还是较好地解决了工业控制装置的可靠性、灵活性、经济性等问题,归纳起来有以下三个方面:

(1) 可靠性高、抗干扰能力强。可编程控制器在耐电磁干扰、耐低温、耐高温、抗潮湿、抗震动等方面均有突出的表现,可靠性高、抗干扰能力强成了可编程控制器最重要的特点之一,被誉为“专为适应恶劣工业环境而设计”的通用控制器。

(2) 功能完善、通用灵活。现代可编程控制器不仅具有逻辑运算、条件控制、计时、计数、步进等控制功能,而且还具有强大的模拟量转换、数据处理以及网络通信等功能,一套系统的输入输出点数可以少至数十点,也可以多至数万点。因此,它既可实现开关量控制,又可进行模拟量控制;既可单机控制,又可以独立或者和其他控制装置共同构成多级分布式控制系统。

(3) 编程简单、使用方便。**梯形图**是可编程控制器第一编程语言,它继承了继电器控制线路的清晰直观感。多数可编程控制器还提供逻辑功能图、指令语言甚至高级语言等编程手段,进一步简化了编程工作,满足不同用户的需要。

此外,可编程控制器还具有接线简单、系统设计周期短、体积小、重量轻、易于实现机电一体化、性价比高等特点,因此在汽车制造、机械、石化、冶金、轻工、电气等很多领域都有广泛的应用。

参考文献

1. 周泽魁. 控制仪表与计算机控制装置. 北京:化学工业出版社,2002

2. 齐蓉,等. 可编程控制器技术. 北京:电子工业出版社,2009

(张光新)

kebiancheng zhidu cunchuqi xinpian

可编程只读存储器芯片(programmable read only memory chip) 一种可由用户自己用电的“硬”编程方法一次性地写入数据的半导体**只读存储器芯片**。简称 PROM 芯片。PROM 芯片产品出厂时是未经编程的,所存储内容是全 0 或全 1。用户可根据自己的使用要求用电的方法编程,将串联于相应存储单元的连接熔丝熔断开路或将相应的连接用 PN 结二极管熔合短路以区别存储内容为 1 或 0。PROM 芯片常称为熔丝 PROM 芯片,该芯片编程是一次性的,一经编程就不能修改存储内容。

早期的 PROM 芯片主要是用快速双极型工艺制作的,1980 年以后 PROM 应用产品中也包括了 CMOS 型。双极 PROM 芯片的连接熔丝有两种类型:一种是使用熔断丝,大多采用铝、镍铬、钛钨、铂硅化物或多晶硅等材料;另一种是使用 PN 结二极管,采用肖特基 TTL 晶片生产工艺,利用扩散方法制造,PN 结熔合短路是由于硅体内使用了一种稳定的硅铝易熔材料。图 1 为 2×2 位 PROM 存储单元阵列,图 1(a)为熔丝熔断开路型,图 1(b)为 PN 结二极管熔合短路型。图中存储单元已被编程写入数据,图 1(a)中右上角熔丝处于开路状态,图 1(b)中除右上角外,其他 3 个二极管处于短路状态。

图 2 为商品 16 kb 寄存器型 PROM 电路框图,其熔丝材料为熔断型多晶硅。该电路由地址寄存器、译码器、128×128 存储阵列和相应输出缓冲器组成,特点是增加了输入锁存地址寄存器。PROM 主要用于微程序控制存储、微处理器程序存储以及查表、字符发生和代码变换等。

随着电子产品的发展,1999 年已实现批量生产的 PPROM 芯片(产品-可编程只读存储器芯片)产品获得了市场的广泛认可。与标准的掩膜 ROM 芯片相比,PPROM 芯片的交货时间降低到掩膜 ROM 芯片的 14% ~33%,但其价格和掩膜 ROM 芯片相当。写入 PPROM 的程序数据不需要使用掩膜板,可以在集成电路封装以后进行,允许在生产的最后时刻改变存储器中的程序数据,因而对掩膜 ROM 技术提出了挑战。具有类似于 PPROM 芯片性能的新型 PROM 芯片产品还有 2003 年实现了批量生产并获得应用的 Matrix 3D 存储器芯片和灵活 ROM 芯片,它们也在抢占掩膜 ROM 芯片的市场份额。

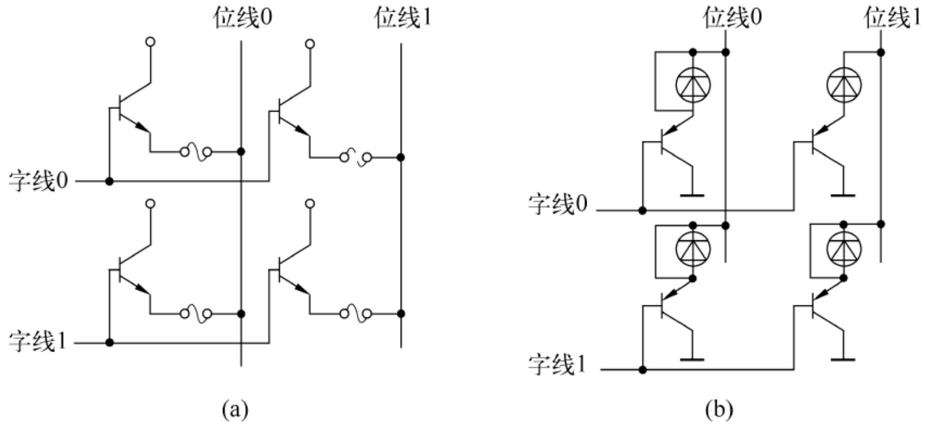


图 1 2×2 位 PROM 存储单元阵列

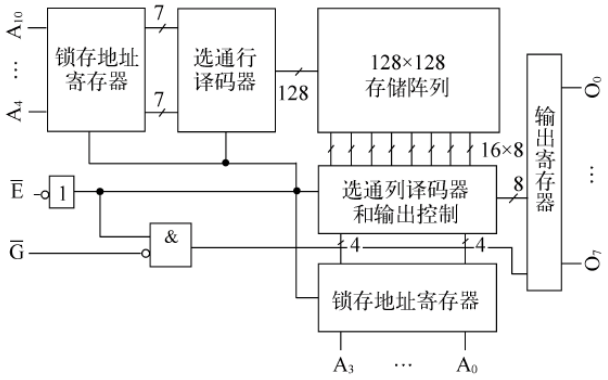


图 2 16kb CMOS PROM 芯片电路框图

参考文献

Metzger L R. A 16K CMOS PROM with polysilicon fusible links. IEEE Journal of Solid State Circuits, 1983, SC-18(5): 562 (时万春)

keca biancheng zhidu cunchuqi xinpian

可擦编程只读存储器芯片(erasable programmable read only memory chip) 一种存储内容可由用户用电的方法写入,并能用紫外线照射的方法擦除后再次写入的半导体**只读存储器芯片**。简称 EPROM 芯片。EPROM 芯片通常用 12 V (或 20 V)的编程引脚在电路系统中写入数据,擦除时需将其从系统中取出并置于紫外线环境下,让光线通过陶瓷封装的石英窗口照射芯片 20～30 分钟,可同时擦除全部原有信息。这种存储器芯片常称为紫外线 EPROM 芯片,记为 UV EPROM 芯片。习惯上还将功能上与 PROM 芯片等同,允许用户使用电的“软”编程的方法写入数据,但不能擦除的只读存储器芯片也归为 EPROM 芯片,称**一次可擦可编程只读存储器芯片**(EPROM 芯片),记为 OTP EPROM 芯片。OTP EPROM 芯片的最原始型式就是将 UV EPROM 芯片装在无石英窗口的塑料封装中。

UV EPROM 芯片是 1971 年开始出现的,早期产品主要使用 NMOS 工艺制造,其存储阵列与只读存储器(ROM)一样,且每个存储单元由一个晶体管组成,用多晶硅浮栅上有无电荷从而影响读出管的输出电平来区分 0/1 状态。由于浮栅周围为绝缘体,电荷可长期保持,所存数据不会丢失。用紫外光可擦除浮栅上的电荷,用高电压可将电荷注入浮栅。这样可实现多次擦除和编程写入。1980 年生产出 16kb CMOS(互补金属-氧化物-半导体)型 UV EPROM 芯片,1986 年已发展到 1 Mb 并逐渐取代 NMOS EPROM 芯片。1990 年集成规模已达 16 Mb, CMOS EPROM 芯片产品已达 EPROM 芯片总量的 84%,成为 EPROM 芯片的主导工艺。

图 1 为典型 1 Mb CMOS EPROM 芯片电路框图。该电路可选择两种字位结构:字宽结构 128 Kb×8 和字宽结构 64 Kb×16。该电路形式的典型写入数据(编程)电压 V_{PP} 为 12.5 V,读数电压 V_{CC} = 5 V,存取时间范围为 100～300 ns。该电路特别增加的输出使能 OE 端可用于消除多重总线系统中的总线竞争。该电路有三个控制引脚,即芯片使能 $\overline{CE}$、编程 $\overline{PGM}$ 和输出使能 $\overline{OE}$。它们的组合可使芯片选择下述状态:读出、输出禁止、芯片待用、编程、编程验证和编程禁止。当 A_9 选择 V_{PP} 时,芯片处于电帖码方式,可以方便地读出生产厂家和器件类型代码。

EPROM 芯片广泛用于办公自动化设备、医疗和通信设备,包括其中的快速语言翻译程序、光学字符识别系统和在线帧存储应用。工业上用于只读存储器的样件开发和工程应用的替换。UV EPROM 芯

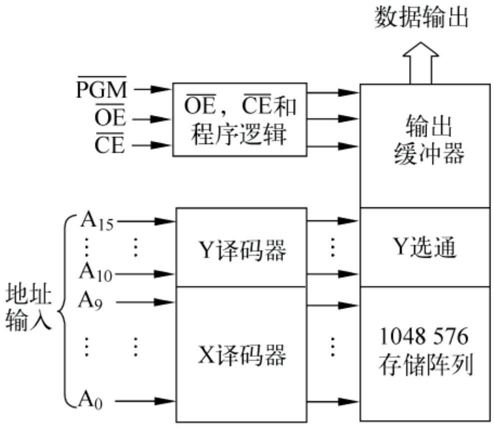


图 1　典型 1 Mb CMOS EPROM 芯片电路框图

片的缺点是成本较高，随存储容量增大其读数性能变坏。OTP EPROM 芯片为塑料封装，成本低，特别适用于快速周转应用，但由于它只能进行一次写入数据（编程）且缺乏测试空白芯片的技术和手段，推广应用有一定局限。

参考文献

Luecke G, Mize J P, Carr W N. Semiconductor memory design and application. New York: McGraw-Hill Book Company, 1973　　（时万春）

keceshixing jishu

可测试性技术（testability technology）　在设计计算机系统时考虑产品的性能可检测的难易程度，在原有设计中插入额外的测试点、逻辑或模块，用于提高产品的可控制性和可观测性的设计技术。可测试性技术的重点在产品的可测试性设计，如何使所插入的测试点、逻辑或模块协调工作而不影响产品的正常功能，以及如何在尽量小的开销下获得较高的故障覆盖率，是可测试性设计的两大主要问题。可测试性设计按设计对象可划分为集成电路可测试性设计、印制电路板可测试性设计、软件可测试性设计等。

在传统的流程中，设计和测试流程是互相分离的，只有在设计流程邻近结束时才会考虑测试。随着计算机系统的复杂程度提高，要对一个系统进行详尽的测试所需要的资源和时间也大为增加。因此在当代设计流程中，设计与测试的融合显著提前，产生了名为可测试性设计的处理流程。提高设计的可测试性可以有效地提高测试覆盖率，降低测试成本，节省测试时间。其中，可测试性包括两个方面：一个是可控制性，即为了能够检测出目的故障或缺陷，可否方便地施加测试向量；另外一个是可观测性，指的是对系统的测试结果是否容易被观测到。需要注意的是可测试性设计时必须要保证在系统正常运行模式下不能对功能产生影响，不能产生附加的活动或者附加的测试。

为了确保设计最大限度拥有可测试性，在开发流程的不同阶段，对于不同的设计对象，需运用特定的可测试性设计技术。常见的集成电路可测试性设计技术包括内建测试（build-in test）、基于扫描路径（scan path）的可测试性设计、边界扫描（boundary scan）技术、稳态电流测试（IDDQ）等。印制电路板可测试性设计包括光板可测试性设计及印制板组装件可测试性设计。其中光板可测试性设计主要考虑布局布线对电气性能及电路通断测试的影响以及测试探头与测试电路物理尺寸匹配问题；印制板组装件可测试性设计主要考虑测试点位置、大小、节距等与测试探头的匹配问题，满足测试的电气条件设置，防止测试对元器件造成损坏等。软件可测试性设计主要手段包括采用测试驱动设计（test driven design）的方法，尽量使函数小型化，使数据的显示与控制分离，坚持可控制性、可分解性、稳定性、易理解性、可观察性、可查询性设计，预留测试驱动和桩的接入点并在增量式开发过程中优先考虑，提供自愈合功能，对于任何操作能产生预期的输出，提供统一的操作执行面板等。

参考文献

雷绍充，等. VLSI 测试方法学和可测性设计. 北京：电子工业出版社，2005　　（尚利宏）

kecexing sheji

可测性设计（design for testability，DFT）　通过在芯片原始设计中插入少量额外电路，以增强电路中间节点的可控制性和可观察性，从而使芯片变得容易测试的方法和过程。可测性设计是芯片设计的重要环节（参见**电子设计自动化**），能大幅度节省芯片测试的成本。可测性设计对原始设计的修改必须在不改变原始设计功能的前提下进行。

主要的可测性设计技术有测试点插入、内部扫描设计、内建自测试、边界扫描设计等。

测试点插入是为了对电路内部某个节点的信号值进行方便的控制或观测，插入的逻辑在非测试方式下不应该改变电路原有的功能。根据测试点的用途，测试点有观测点和控制点之分。观测点的插入通常不影响数据线的功能，但可能会产生一定的延

迟。插入控制点时,通过多路选择器把原功能数据线断开,从而可在测试方式下选择来自芯片测试输入引脚的数据。控制点的插入增加了多路选择器的硬件逻辑和测试控制引脚,增加了多路选择器选通的延迟,且测试方式下电路不能实施某些正常功能。

内部扫描设计本质上也是一种测试点插入技术。时序电路中的记忆元件(如触发器)的取值体现了电路的状态,由于很难控制或观测它们,所以很难对时序电路进行测试。内部扫描设计就是把这些记忆元件修改为扫描触发器,并连接成为扫描链(相当于移位寄存器)。扫描链的输入在芯片引脚可控,其输出在引脚可观测,从而达到对记忆元件的取值进行控制和观测的目的。这样一来,进行内部扫描设计后的时序电路的测试可以参照组合电路测试的方式来进行。

内建自测试(BIST)相当于把"测试仪"做到了电路内部,因为它既要对被测电路提供输入的测试向量,又要将其输出结果与期望的结果进行比较以判定测试是否通过。由于这个"测试仪"专门为这个被测电路而设计,所以其功能单一固定。根据需要,BIST 用于测试的电路可以是一个数字逻辑电路(这时 BIST 称为逻辑 BIST),也可以是存储器(这时 BIST 称为存储器 BIST)或其他模拟电路。BIST 减少了测试工作对自动测试仪(ATE)的依赖性,可为高速电路提供在电路正常工作时钟频率下的测试,甚至还支持在线测试。这些特点减少了测试成本,并且有助于提高系统的可靠性和可用性。

边界扫描设计的目的是在电路板一级对芯片或板上逻辑与连接进行测试、复位和系统调试。在电路板一级,一般不能对芯片的引脚直接进行控制或观测。通过边界扫描,可以对芯片引脚与核心逻辑之间的连接进行扫描,也就是通过串行移位进行控制或观测。边界扫描设计要遵循 IEEE 1149.1 标准定义的国际上通用的芯片边界扫描结构及其测试访问端口规范。

参考文献

1. Bushnell M L, Agrawal V D. Essential of electronic testing for digital memory and mixed-signal VLSI circuits. Kluwer Academic Publishers, 2000

2. 李华伟,鲁巍. 数字系统测试与可测试设计. 北京:机械工业出版社,2006

(李晓维)

kechonggou jisuanji

可重构计算机(reconfigurable computer)

结合**通用计算机**编程的灵活性和定制硬件结构的高性能,兼顾两者优点的**计算机**。可重构计算机可以通过改变硬件结构中的数据通路或不可见的连接方式改变计算机的结构,从而使得计算机的结构更适合于特定的计算任务,或者在运行过程中通过适时地以硬件连接的方式调用特别的功能部件使得计算机的结构更适合于特定的计算任务。与通用计算机相比它具有改变自身结构的能力,适应各种应用的特殊性,从而提高计算性能;与定制的硬件相比它能够在运行时通过软件改变电路结构,具备更好的灵活性。

可重构计算的概念始于 20 世纪 60 年代, Gerald Estrin 首先提到了与通用处理器耦合在一起的可以重构的硬件阵列。通用处理器可以控制硬件功能变换,完成多种特定任务,同时其性能可以达到定制硬件的速度。这一思想描述了一种具有硬件结构重新构造的混合式计算机体系结构。但是,限于当时的硬件条件,这一思想没有具体实现。可重构计算机研究在 20 世纪 80~90 年代快速发展,但是独立的商品化的可重构计算机并没有实现。

可重构计算机根据重新构造的时间分为静态重构和动态重构。静态重构是指在系统运行之前进行电路功能配置;动态重构则是指在系统运行期间动态配置电路功能,动态配置过程中还可以进行部分电路配置,也称为部分可重构,增加了系统重新构造的灵活性。可重构计算机还可以根据重新构造的粒度分为粗粒度可重构和细粒度可重构。粗粒度是指系统可重新构造的运算单元的位宽在字节以上,同时支持多种算术逻辑运算;细粒度是指可重构基本运算单元的位宽是"位"(bit),支持位的逻辑运算。

以 FPGA(field programmable gate array)为代表的细粒度可重构计算是目前应用最广泛、商业化程度最高的可重构计算形式。FPGA 是一种电子器件,在制造后仍然可以被用户或设计师配置其功能,因而称为现场可编程门阵列器件。它能够完成按应用需求定制的集成电路(ASIC)芯片同样的逻辑功能,同时具备根据用户需要变换逻辑功能的能力,因此被称为**可重构器件**。FPGA 在结构上主要包括两大部分,一部分是可编程逻辑单元,也称为逻辑块,逻辑块之间通过层次式的可配置互连线连接在一起。逻辑块可以通过配置完成多种复杂的组合逻辑功能或者是最简单的"与""或""非"逻辑功能。通过配置可以设置互连线的导通或截断。典型 FPGA 逻辑块由一个 4 输入或 6 输入查找表(LUT)和一个触发器组成,其中查找表用于实现基本逻辑功能。FPGA 用存储逻辑功能和存储连线关系的方法实现

了细粒度可重构思想。

基于 FPGA 的细粒度可重构计算在定制计算结构的基础上通过配置可以实现任意逻辑功能,因此具有实现任意功能的灵活性,避免了 ASIC 功能固定的缺点,同时能够将计算过程映射到一个高效的定制计算结构上,提高计算效率。FPGA 在具有可重构这种灵活性的同时,也会带来副作用,例如,为了保持任意逻辑块之间能够有足够多的通道用于连接,与 ASIC 芯片相比,FPGA 很大比例的芯片面积用于布置各种长短连线和交叉开关;同时因为这些预先布置的连接线数量有限,在实际布线的过程中,逻辑块之间可能经过多次转接,导致线延迟增大。

在编程模式方面,粗粒度可重构计算结构都是在传统计算模式上,通过改变运算单元的连接关系达到可重构的目的,因此都采用高级语言开发应用程序,但是需要手工描述运算单元之间的连接关系,工作量大,难度高。FPGA 通常采用寄存器传输级的设计方法,效果较好,但是需要专业人员才能掌握。人们仍然在探索采用高级语言编程,得到高效的 FPGA 定制电路结构。

在可重构计算机研究方面,具有可重构能力的计算部件将与通用处理器进一步融合。随着单芯片集成晶体管容量的增加,可以在处理器内部集成 FPGA 可重构计算单元。另一方面,FPGA 芯片的功能会更加丰富或系列化,在 FPGA 提供灵活的逻辑功能基础上,会提供更多用户需要的定制功能单元,比如面向信号处理应用的大量 DSP 运算模块、浮点运算单元或者高性能的处理器。

参考文献

1. Hauck S, Fry T W, Hosler M M, Kao J P. The Chimaera reconfigurable functional unit. IEEE Transactions on Very Large Scale Integration Systems, 2004, 12(2) February: 206-217

2. Mei Bingfeng, Vernalde S, Verkest D, Lauwereins R. Design methodology for a tightly coupled VLIW/reconfigurable matrix architecture: a case study. Proceedings of the conference on Design. Automation and Test in Europe (DATE 2004). Paris, France: 2004, 21-24

3. Baumgarte V, Ehlers G, May F, et al. PACT XPP—a self-reconfigurable data processing architecture. The Journal of Supercomputing. 2003, 26(2): 167-184

(实勇)

kechuandai jisuan

可穿戴计算(wearable computing)　将**计算机**穿戴在用户身上进行信息处理和通信的技术。用这种技术所研制的计算机称为**可穿戴计算机**。用户可在工作或日常活动中使用这种计算机,并且可以持续地和它进行交互。

可穿戴计算的概念早在 20 世纪 50 年代就出现了,但相关技术的研究与应用是在 90 年代以后才随着电子、计算机和通信技术的发展而迅速发展起来的。目前可穿戴计算机的应用正逐步从军事向更广阔的领域拓展,如设备安装、采掘工业、野外勘探、医疗监测、新闻采访,并已经走进普通人的日常生活,如健康监护。

可穿戴计算不仅将计算机微型化并与服饰融为一体,而且实现了人机的紧密结合,体现了以人为本的理念。可穿戴计算机一般配有头戴式或眼镜式超微型显示器,主机和其他外设嵌在衣物中,采用多模态接口和多通道传感技术,特别适合在户外和移动中使用。虽然不同用途的可穿戴计算机在构成、功能、形态等方面很不相同,但有其共同的特征,包括:①人机一体　作为一种最自然的携带方式,可穿戴是其本质,需要在人机工程学的指导下,通过片上系统(SOC)体系结构、**嵌入式操作系统**等技术使计算机主机微型化,并根据应用的需要使其具有多端口和高性能 I/O;②操作便捷　人机自然交互是其可用的关键,桌面计算中键盘、鼠标输入的方式已不适用,最终需要通过多模态接口、多通道传感技术和情境感知(context awareness)技术捕获人的行为语义和自然表达作为交互请求,目前单手键盘是一种最可靠而且实用的操作模式;③移动中使用　用户可在移动状态下使用是其突出特点,要求能够适应人的活动并且作为一个移动节点随时联网,需要建立以人为节点的移动计算系统;④持续工作　可持续性是其区别于其他移动设备的重要特征,总是处于工作状态的计算机还可通过情境感知等技术主动感知用户的环境并自动做出响应。

可穿戴计算目前已经形成一个交叉研究领域,需要解决的一些特殊的关键技术有:①片上系统体系结构　把计算机主机的硬件集成到一个低功耗、高性能的芯片里;②多端口、高性能 I/O　用微小型的主机连接许多外设;③嵌入式操作系统　实时的和微内核的操作系统,并具有强大的处理多外围设备的能力;④外围设备的设计　所有外设需要是高性能、小体积、低功耗,符合人体特征,有利健康,

安全可靠的;⑤无线连接技术　分布于身体各部位的多个部件之间以近距离无线通信取代连线;⑥无线自组网络　没有固定路由器支持的各节点以任意方式移动并动态连接;⑦人机交互　解决用户与计算机之间的交互问题以及支持用户对环境的感知能力,包括听觉和视觉方面的识别技术、传感技术、信息融合技术和情境感知技术;⑧移动数据库技术　支持用户在移动时的网络断接频繁、条件多样、信道不对称,计算部件伸缩范围大等条件下的数据库管理技术;⑨电子织物(electronic textiles 或 smart textiles)　嵌入了计算和通信用电子部件的织物。

可穿戴计算机在不断更新换代,可以预见的是:①由于专用芯片和专用设备的改进,可穿戴计算机将越来越小,交互技术的提高使得使用也越来越方便;②穿戴方式将有很大的改进,可以放在手表里、眼镜上甚至植入皮下;③将出现许多可选的专用设备和软件,使系统的构成能做到随心所欲;④可穿戴计算机的外设之间将实现无线互连。　(史元春)

kejisuan hanshu

可计算函数(computable function)　能够在抽象计算机上计算其值的函数。1936年,几位数理逻辑学家从不同角度给出了可计算函数的精确描述,并且证明,这些可计算函数类相同。J. C. Shepherdon 和 H. E. Sturgis 于1963年给出了无限寄存器机器(URM)。用URM定义的可计算函数类与用其他方法定义的可计算函数类相同。URM模型比图灵机模型易于理解。下面介绍URM可计算模型。

URM有无穷多寄存器 $R_1, R_2, R_3, \cdots$。任何时刻,每个寄存器均含一个自然数,用 r_n 表示 R_n 中的自然数。寄存器的内容可以通过指令改变。有穷多条指令构成一程序。URM指令有4种(下面的 N 为自然数集合):

(1) 清零指令 $Z(n), n \in N, Z(n)$ 将 R_n 的内容变为0,其余不变。

(2) 后继指令 $S(n), n \in N, Z(n)$ 将 R_n 的内容加1,其余不变。

(3) 传输指令 $T(m,n), m,n \in N, T(m,n)$ 将 R_m 的内容 r_m 移入 R_n 中,其余不变(包括 R_m 也不变)。

(4) 跳转指令 $J(m,n,q), m,n,q \in N$,在程序 P 中遇到指令 $J(m,n,q)$ 时就比较 R_m, R_n 的内容,如果 $r_m = r_n$,则URM进行 P 的第 q 条指令。如果 $r_m \neq r_n$,则URM依次执行下一条指令。如果因 P 少于 q 条指令而不可能跳转,则URM停止运算。

要URM执行一计算,必须给出程序 P 并设置初态。所谓初态,指寄存器中初始自然数序列。

假设 P 是一个URM程序, $a_1, \cdots, a_n, b \in N$,记号 $P(a_1, \cdots, a_n, 0, 0, \cdots)$(简记为 $P(a_1, \cdots, a_n)$)表示将程序 P 作用于寄存器初态为 $a_1, \cdots, a_n, 0, 0, \cdots$ 的自然数序列。如果计算 $P(a_1, \cdots, a_n)$ 终将停止,停止时, R_1 中内容为 b,则称 $P(a_1, \cdots, a_n)$ 收敛于 b,记为 $P(a_1, \cdots, a_n) \downarrow b$。设 f 是一个从 N^n 到 N 的部分函数,如果对任意 $a_1, \cdots, a_n, b \in N$, $P(a_1, \cdots, a_n) \downarrow b$ 当且仅当 $(a_1, \cdots, a_n) \in \mathrm{dom}(f)$ 且 $f(a_1, \cdots, a_n) = b$,则称程序 P 计算了 f。

如果存在一个URM程序计算 f,则称函数 f 是URM可计算的。

例如,函数 $x+y$ 可用URM程序

I 1　J(3,2,5)
I 2　S(1)
I 3　S(3)
I 4　J(1,1,1)

来计算。

寄存器的初态为: $x, y, 0, 0, \cdots$;程序不断地将1加入 R_1 中,用 R_3 来记录加1的次数,一旦加1次数为 y,则停机,此时, R_1 中的数恰好为 $x+y$。因此, $x+y$ 是URM可计算的。

函数

$$f(x) = \begin{cases} x/2, & \text{如果 } x \text{ 是偶数} \\ \text{无定义}, & \text{如果 } x \text{ 是奇数} \end{cases}$$

是URM可计算的。令初态为 $x, 0, 0, \cdots$ 执行程序如下:

I 1　J(1,2,6)
I 2　S(3)
I 3　S(2)
I 4　S(2)
I 5　J(1,1,1)
I 6　T(3,1)

其中,当I6执行完后,因无后续指令,自行终止。

20世纪30年代,可计算函数概念的出现,标志着一个新的数学分支的诞生,它在计算机科学、数学、哲学等领域均有重要的意义。

参考文献

Cutland N. Computability, an introduction to recursive function theory. Cambridge, UK: Cambridge University Press, 1980

(陈火旺　贲可荣)

kejisuanxing lilun

可计算性理论(computability theory)　研

究计算的一般性质的数学理论。计算的过程就是执行**算法**的过程。可计算性理论的中心课题就是将算法这一直观概念精确化,建立计算的数学模型,研究哪些是可计算的,哪些是不可计算的,以此揭示计算的实质。由于计算是与算法联系在一起的,因此,可计算性理论也称算法理论。

直观上说,求解一类问题的算法是一组规则。这组规则条数有限,每一条都是可执行的(可操作的),并且这种操作性是绝对机械的,即不论何人、何时对之进行操作,只要输入数据相同,其结果都是一样的(唯一确定的)。作为算法的一组规则,至少还应包含一条有关终止计算的条目。因此,直观上算法所具备的特征为:有限性、可执行性、机械性、确定性和终止性。求解一类问题的算法的执行过程就是从问题的初始数据开始,依次执行每条规则,直至终止。这个过程是有限的,在数理逻辑中,称这样的算法执行过程是一个"能行"过程,并称可计算性理论为**能行性理论**。如果对终止性不加要求,这类算法是不完全的,或称是部分的,即通常所说的过程。

例如,求两个正整数 m,n(设 $m\geqslant n$)最大公因子的欧几里得算法(辗转相除法),可用下述三条规划描述:

R_1[求余数]:以 n 除 m 得余数 r;

R_2[测试]:若 $r=0$,终止计算,n 即为答案;否则转到步骤 R_3;

R_3[互换]:变 m 的值为 n,变 n 的值为 r,转到步骤 R_1。

依照这三条规则指示的步骤,可在有限步里计算出任何两个正整数的最大公因子。计算的过程就是执行这三条规则的步骤所构成的序列。显然,这个过程是确定的和有限的。

多少年来"算法"、"计算"这些概念似乎并不存在什么问题。到了 20 世纪 20 年代,数学家们对此提出了疑问,到底计算的实质是什么?于是开始了对算法的概念精确化的研究。算法概念精确化的途径很多,其中之一是形式地定义抽象计算机,把算法看作是抽象计算机的程序。进而把那些存在算法计算其值的"可计算问题"(或"可解问题")精确定义为:能够在抽象计算机上编出程序计算其值的问题。在这种抽象计算机计算模型的基础上,就可以从理论上讨论哪些是可计算的,哪些是不可计算的。对于函数,凡存在抽象计算机程序计算其值者称为可计算函数。对于一般数学问题,凡存在抽象计算机程序求解者称为可解的问题。函数值的计算和问题的求解这两者是可以互相转化的。例如,在自然数论域里,询问问题"数 n 是否为素数"是否可解的,等于询问函数

$$f(n)=\begin{cases}1, & n\text{ 是素数}\\0, & n\text{ 不是素数}\end{cases}$$

是否可计算的。

可计算理论起源于对数学基础问题的研究。从 20 世纪 30 年代开始,为了讨论所有问题是否都有求解的算法,数学家和逻辑学家们从不同角度提出了几种不同的算法精确化定义。为简洁起见,许多数学家都起始于对自然数论域里的数论函数的可计算性的研究,提出了几种可计算函数定义。K. Godel,J. Herbrand 和 S. C. Kleene 于 1936 年定义了递归函数;A. Church 于 1935 年提出了 λ 转换演算;A. Turing 和 E. Post 分别于 1936 年和 1943 年提出了各自的抽象计算机模型(后人把 A. Turing 提出的抽象机称为图灵机);A. A. Mapkob 于 1951 年定义了正规算法;J. C. Shepherdson 于 1963 年定义无限寄存器机器;50 年代末和 60 年代初,胡世华和 J. McCarthy 等人各自独立地提出了定义在字符串上的递归函数;如此等等。后来陆续证明了,上述这些不同计算模型(算法精确化定义模式)的计算能力都是一样的,它们所刻画的函数类均相同,即它们是等价的。

可计算性理论的主要内容包括以下诸方面。

图灵机　一种在理论计算机科学中广泛采用的抽象计算机,A. Turing 于 1936 年提出,用于精确描述算法的特征。可用一个图灵机来计算其值的函数是可计算函数,找不到图灵机来计算其值的函数是不可计算函数。可以证明,存在一个图灵机 U,它可以模拟任何其他的图灵机。这样的图灵机 U 称为通用图灵机。通用图灵机正是后来的存储指令的通用数字计算机的理论原型。

λ 转换演算　一种定义函数的形式演算系统,是 A. Church 于 1935 年为精确定义可计算性而提出的。他引进 λ 记号以明确区分函数和函数值,并把函数值的计算归结为按一定规则进行一系列转换,最后得到函数值。按照 λ 转换演算能够得到函数值的函数称为 λ 可定义函数。

丘奇-图灵论题　可计算性理论的基本论题,也称图灵论题。它规定了直观可计算函数的精确含义。丘奇论题说:λ 可定义函数类与直观可计算函数类相同。图灵论题说:图灵机可计算函数类与直

观可计算函数类相同。A. Turing 证明了图灵机可计算函数类与 λ 可定义函数类相同。这表明图灵论题和丘奇论题讲的是一回事,因此把它们统称为丘奇-图灵论题。直观可计算函数不是一个精确的数学概念,因此,丘奇-图灵论题是不能加以证明的。20 世纪 30 年代以来,人们提出了许多不同的计算模型来精确刻画可计算性,并且证明了这些模型都与图灵机等价。这表明图灵机和其他等价的模型确实合理地定义了可计算性,因此丘奇-图灵论题得到了计算机科学界和数学界的公认。

原始递归函数　自变量值和函数值都是自然数的函数,称为数论函数。原始递归函数是数论函数的一部分。首先规定少量显然直观可计算的函数为原始递归函数,它们是:函数值恒等于 0 的零函数 C_0,函数值等于自变量值加 1 的后继函数 S,函数值等于第 i 个自变量值的 n 元投影函数 $P_i^{(n)}$。然后规定,原始递归函数的合成仍是原始递归函数,可以由已知原始递归函数简单递归地计算出函数值的函数仍是原始递归函数。例如,和函数 $f(x,y)=x+y$ 可由原始递归函数 $P_1^{(1)}$ 和 S 递归地计算出函数值

$$\begin{cases} f(x,0) = P_1^{(1)}(x) \\ f(x,S(y)) = S(f(x,y)) \end{cases}$$

比如,$f(4,2)$ 可这样计算,首先算出 $f(4,0)=P_1^{(1)}(4)=4$,然后计算 $f(4,1)=S(f(4,0))=S(4)=5$,最后 $f(4,2)=S(f(4,1))=S(5)=6$。因此,和函数是原始递归函数。显然,一切原始递归函数都是直观可计算的。许多常用的处处有定义的函数都是原始递归函数,但并非一切直观可计算的、处处有定义的函数都是原始递归函数。

部分递归函数　为了包括所有的直观可计算函数,需要把原始递归函数类扩充为部分递归函数类。设 $g(x_1,\cdots,x_n,z)$ 是原始递归函数,如果存在自然数 z 使 $g(x_1,\cdots,x_n,z)=0$,就取 $f(x_1,\cdots,x_n)$ 值为满足 $g(x_1,\cdots,x_n,z)=0$ 的最小的自然数 z;如果不存在使 $g(x_1,\cdots,x_n,z)=0$ 的自然数 z,就称 $f(x_1,\cdots,x_n)$ 无定义。把如上定义的函数 f 加到原始递归函数类中,就得部分递归函数类。因为不能保证如上定义的 f 在一切点都有定义,故称其为部分函数。处处有定义的部分递归函数称为全递归函数。部分递归函数类与图灵机可计算函数类相同。对于每个 n 元部分递归函数 f,可以编一个计算机程序 P,以自然数 $x_1,\cdots,x_n$ 作为输入,若 $f(x_1,\cdots,x_n)$ 有定义,则 P 执行终止并输出 $f(x_1,\cdots,x_n)$ 的值,否则 P 不终止。

递归集　递归集最初是针对自然数的子集定义的,它们是存在算法判定每个自然数是否为其元素的集合。可以将递归集的概念推广到其他集合。所讨论的对象的全体称为域,如果存在算法判定域中任意元素是否属于 A,则称 A 为递归集。对于每个递归集,可以编一个计算机程序,以域中任意元素作为输入,计算机执行该程序都可给出适当的输出,表明该元素是否属于这个集合。有许多集合不是递归集。

递归可枚举集　如果对于集合 A 可以编一个程序 P,输入域中任意元素 x,若 $x \in A$,则 P 的执行将终止并输出"是",否则 P 的执行不终止,就称 A 为递归可枚举集。A 为递归可枚举集的充分必要条件是可以编一个程序枚举 A 的元素,即打印 A 的元素,使得对于 A 中任意元素,只要时间足够长总会在打印纸上出现。递归集都是递归可枚举集,但有些递归可枚举集不是递归集。有许多集合不是递归可枚举集。

可判定性和半可判定性　判定问题是无穷多个同类个别问题的总称。例如,2 是不是素数? 6 是不是素数? 这些都是个别问题,把这类个别问题概括起来,就得到一个判定问题:任意给定的正整数是不是素数? 这里的正整数集合称为该判定问题的域,给定域中的一个元素,判定问题就对应一个个别问题。对于一个判定问题,如果能够编出一个程序,以域中任意元素作为输入,执行该程序就能给出相应的个别问题的答案,就称该判定问题为可判定的(可解的)。例如,"任意正整数是不是素数"这个问题就是可判定的。对于集合 A,域中任意元素是否属于它的问题称为集合 A 对应的判定问题。集合是递归集的充分必要条件为对应的判定问题是可判定的。因此,全体素数的集合是递归集。

对于一个判定问题,如果能够编出一个程序 P,以域中任意元素作为输入,当相应的个别问题的解答是肯定的时候,P 的执行将终止并输出"是",否则 P 的执行不终止,就称该判定问题为半可判定的(半可解的)。可判定的问题总是半可判定的。集合是递归可枚举集的充分必要条件为对应的判定问题是半可判定的。

图灵在 1936 年证明,图灵机的停机问题是不可判定的,即不存在一个图灵机能够判定任意图灵机对于任意输入是否停机。图灵机的停机问题是半可判定的。图灵机的停机问题是很重要的,由它可以推出计算机科学、数学、逻辑学中的许多问题,如可

以证明希尔伯特第10问题(刁番图方程是否有整数解的问题)是不可判定的。其证明也借助于停机问题的不可判定性。

可计算性理论是计算机科学的理论基础之一。早在20世纪30年代,图灵对存在通用图灵机的逻辑证明表明,制造出能编程序来作出任何计算的通用计算机是可能的,这影响了40年代出现的存储程序计算机(即冯·诺依曼型计算机)的设计思想。可计算性理论确定了哪些问题可能用计算机解决,哪些问题不可能用计算机解决。例如,图灵机的停机问题是不可判定的表明,不可能用一个程序来判定任意程序的执行是否终止,避免了人们为试图编制这样的判定程序而无谓地浪费精力。

可计算性理论中的基本思想、概念和方法,被广泛应用于计算机科学的各个领域。建立数学模型的方法在计算机科学中被广泛采用。递归的思想被用于程序设计,产生了递归过程和递归数据结构,也影响了计算机的体系结构。λ演算被用于研究程序设计语言的语义,例如,表处理语言就以λ转换演算为理论基础。

递归函数论的建立对于数学基础的研究具有十分重要的作用。为了证明不完全性定理,K. Godel发明了原始递归函数。20世纪初最伟大的数学家希尔伯特曾希望将整个数学形式化,建立了一个协调、完全的大系统。哥德尔不完全性定理表明,只要表达能力足够丰富,这种形式系统就不可能是完全的。例如,这种系统的协调性问题自身就无法在此系统内部得到证明。K. Godel的发现对数学具有重要的影响,对认识论乃至整个哲学也有深刻的意义。

参考文献

1. Rogers H. Theory of recursive functions and effective computability. New York:McGraw-Hill,1967

2. Cutland N. Computability, an introduction to recursive function theory. Cambridge, UK: Cambridge University Press,1980　(陈火旺　何自强　贲可荣)

kekaoxing sheji

可靠性设计(reliability design)　研究计算机系统在设计过程中达到可靠性指标要求的设计技术。**可靠性**是计算机系统在规定的条件下和规定的时间内,完成规定功能的能力。规定的条件通常包括环境条件、使用条件和维修条件和操作技术,规定的时间是所需考虑的时间范围,规定的功能是指计算机的各项技术性能指标和失效判据。能力是在规定的条件下和规定的时间内,完成规定功能的程度。可靠性指标可用可靠度或平均故障间隔时间定量化表示(参见**计算机硬件可靠性**)。可靠性设计是计算机工程设计的一个重要方面。在进行基本性能设计、经济性设计的同时,要进行可靠性设计。也就是从系统的总体设计、原材料、元器件选用、电路设计、结构工艺及可维性设计等各方面综合考虑,尽量挖掘各方面潜力,采取降额、冗余等各种措施来实现系统所要达到的可靠性指标要求。

可靠性设计一般遵循如下原则:

(1) 在满足功能和性能指标的前提下,把可靠性放在第一位。

(2) 在满足功能和性能要求的前提下,尽可能简化设计,采用简单的、少量的部件、器件、接口和电源等,以降低功耗、体积、重量和失效率,提高可靠性水平。

(3) 在减少数量的基础上尽量压缩品种、规格,以利于供应、质量监督、筛选测试和调试维修。

(4) 贯彻设计的标准化、系列化和模块化原则,以利于系统扩充和各型号间的兼容性。

(5) 对性能、可靠性、成本等要综合考虑。

可靠性设计内容包括:①建立系统及各分系统的可靠性模型;②可靠性预计;③可靠性指标分配。可靠性设计通常不是一次就可全部完成,要经过预计、分配、再预计、再分配的反复过程;而且在研制阶段基本结束后,还要经过各种可靠性试验,对所暴露出的缺陷采取必要的措施,使可靠性进一步提高,最后才能将可靠性设计定型。

可靠性模型　可靠性模型以产品的功能框图、原理图和工程图为基础,建立系统、分系统或设备的可靠性框图,即产品的可靠性物理模型。在框图中方框表示评定系统可靠性时必须考虑的单元,所有连接方框的线不考虑可靠性值,即可靠性等于1。产品各单元间连接用的导线或连接器的可靠性必须考虑时,可单独放入一个方框或作为某个单元的一部分。所有方框的失效率是互相独立的,与它们的排列顺序无关。

找出各方框的可靠性数学表达式及各方框之间的可靠性关系,建立以各方框的可靠性为基础的系统、分系统或设备的可靠性表达式,即可靠性模型。其目的是确定产品成功或失败的概率与各单元成功或失败概率的关系。

以某计算机为例,它是由中央处理器(CPU)板、内存板和输入输出(I/O)板及电源和机箱5个部分

组成的无冗余简单串联系统,图 1 是其可靠性框图。



图 1　某计算机可靠性框图

若每个部件的可靠性为 R_i,则总的可靠性 R_T 为

$$R_T = \prod_{i=1}^{5} R_i \qquad (1)$$

各部件的失效率 λ_i 是个常数,可靠性 $R_i = \exp(-\lambda_i t)$,则总可靠性为

$$R_T = \prod_{i=1}^{5} R_i = \exp\left(-t\sum_{i=1}^{5}\lambda_i\right) = \exp(-\lambda_T t) \qquad (2)$$

式中,t——工作时间;

λ_T——产品所有部件失效率的总和。

可靠性预计　其目的是根据可靠性框图和数学模型预计计算机系统、分系统或设备的可靠性,并确定所提供的设计方案是否达到了任务书或合同规定的可靠性要求,为确定设计方案和选择元器件、材料等提供依据。可靠性预计工作的基本要点是:①应对计算机系统、分系统或设备做出可靠性预计,为寿命周期费用、后勤保障分析、产品的应用效能分析提供依据。特别要根据预计中的数据和结果指出失效概率较高的设备,为设计修改、故障分析提供依据。②在方案构思阶段和正式设计阶段采用不同的方法进行可靠性预计。③随着设计、元器件的选择和使用环境的改变,可靠性预计工作应及时修改,使之能反映产品的真实水平。

在方案构思阶段的可靠性预计是用比较有限的资料较迅速地做出可靠性预计,其目的是对方案的合理性、可行性作出概略的估计与判断。此种预计法无须进行详细的分析和精确的计算,通常采用元器件计数法。该方法需要三方面数据:①设备所有元器件的种类及数量;②拟采用的各种元器件的质量等级;③设备的使用环境。

正式设计阶段的可靠性预计通常采用元器件应力分析法,这种方法需要大量的详细信息,尤其是元器件各种应力在内的详细数据,其设计精度比较高,可以满足工程设计的精确性要求。

有关元器件计数预计法和元器件应力分析预计法的详细情况参见 GJB 299—1987《电子设备可靠性预计手册》。

可靠性指标分配　从系统或整机的角度出发,根据经过论证确定的系统可靠性指标要求,按照一定的原则和方法分配给各分系统,进而再将各分系统的可靠性指标分配给各个部件、电路,直至元器件、印制板、接插件等。可靠性指标分配要确保分配的合理性(协调性)、技术上的可行性和资源利用的经济性。可靠性指标分配的方法有许多种,具体采用哪一种,取决于所构成的系统模型,已掌握的数据材料,现有技术条件以及体积、重量、功耗等限制条件。可靠性指标分配应反复进行,当某一分系统所分配的指标技术上不可行时,应修改方案,进行重新分配,直至满足设计要求为止。主要的可靠性指标分配方法有:考虑重要程度和复杂程度的代数分配法,考虑复杂程度的分配法,各分系统失效概率可预计情况下的分配方法,顺序分配法,按经验比率分配法等。

可靠性设计软件　可靠性设计是一项复杂的系统工程,为了科学、准确、高效地做好此项工作,出现了可靠性设计软件,利用计算机,在电子设计等软件的配合下进行可靠性设计。这类软件可进行可靠性预计分析、可靠性框图分析、失效数据分析、故障模式、影响及危害度分析、故障树分析、寿命周期费用分析,发现设计中的薄弱环节,为设计改进或生产过程控制提供依据。

参考文献

1. 傅佩琛,赵霖,张军英. 计算机系统硬件软件可靠性理论及其应用. 北京:国防工业出版社,1990

2. Siewiorek D P, Swarz R S. 可靠系统的设计理论和实践. 袁由光,曹泽翰,刘志模,等译. 北京:科学出版社,1988

(刘恩德)

kekuochong yuyan

可扩充语言(extensible language)　具有扩充机制的**程序设计语言**。一个程序设计语言称为可扩充的是指用户能根据自己的需要,利用该语言提供的扩充手段,往该语言中增加新的成分或新的功能,而又无须改动该语言的编译程序。

扩充机制就性质分,有词法扩充、语法扩充和语义扩充等层次。例如,关键词和运算符换名,通过宏展开引进新的文字等属于词法扩充。定义复合数据类型,引进新的运算符等属于语法扩充。确定新运算符的运算规则(如结合律、交换律)和原有运算符的重载(如改变其定义域)等属于语义扩充。

历史上把可扩充性作为特色加以强调的语言有 EL1(相应的实现系统为 ECL),ALGOL 68,FORTH,SIMULA 67 等。其中 SIMULA 67 可以在系统类的基

础上插入新的类说明，从而把它扩充为各种专用语言。这种扩充方式一直沿用至今。

可扩充语言的研究兴起于 20 世纪 60 年代中期。当时，日益扩大的计算机应用领域向计算机语言提出了各种功能要求。如果把这些功能都纳入一个计算机语言中便会使语言的规模过度膨胀。由此人们企图以适度规模的语言加上可扩充功能来适应这种需要。美国哈佛大学的 T. E. Cheatham 等人是研究可扩充语言的带头人。该领域在 70 年代初成为研究热点，到 70 年代中期，便开始降温。著名程序设计语言评论家 J. Sammet 在 1981 年指出："可扩充语言作为研究方向已经死亡"。其原因是：程序设计语言的研究在 70 年代经历了一场从追求功能丰富到追求程序可靠的变革。此外，可扩充语言的研究并未像人们预期的那样给程序设计带来革命性的变化。但是，有关的研究成果仍然体现在当代的一些重要程序设计语言中。（陆汝钤）

ke panding wenti

可判定问题（decidable problem） 具有求解算法的问题。判定问题的一系列研究结果是 20 世纪最重大的学术成就之一。Hilbert 1900 年在国际数学家大会上提出的 23 个问题中的第 10 个问题，即丢番图方程解的存在性判定问题就是一个著名的判定问题。另一个著名的判定问题是所谓的数理逻辑基本问题：在一阶逻辑中，是否存在判别任意命题为定理的一般方法。正是对这些问题的研究，直接推动了递归论的产生。在 1936 年，S. Kleene 提出了递归函数的概念，A. Turing 提出了图灵机模型，A. Church 和 A. Turing 独立地证明了数理逻辑的基本问题是不可判定的。

由于各种形式的判定问题都可转化成某一集合 A 的从属关系的判定问题：是否存在算法能判定任意 x 是否属于集合 A。因此，可判定问题可用下述形式定义：若 A 是递归集，则称 A 的从属关系的判定问题是可判定的，或称递归可解的。

由于绝大多数著名的判定问题都是不可判定的，所以通常研究一个问题的可判定性一般是指证明它的不可判定性。

参考文献

1. Rogers H. Theory of recursive functions and effective computability. New York：McGraw-Hill 1967

2. 莫绍揆. 递归论. 北京：科学出版社，1987

（马绍汉　李大兴）

kepeizhi chuliqi

可配置处理器（configurable processor） 一种内部的硬件结构、功能模块、外部接口、指令集等可根据应用需求灵活调整、裁剪的处理器。其与可重构计算（reconfigurable computing）中的可重构处理器不同，后者指的是在计算运行过程中，处理器可以根据不同的任务进行硬件结构重构（往往通过现场可编程门阵列实现）以适应计算需求，提升处理能力。而可配置处理器通常指的是在设计阶段，针对应用需求，采用硬件结构调整、接口配置、指令增删的方法，从处理器的初始配置开始将其调整为适用于目标指令的优化处理器，并进行后续设计以及生产。

设计理念

可配置处理器的设计理念是以应用为导向，自顶向下进行处理器结构设计。而现有的通用处理器、通用**图形处理器**则不同，需要应用为它们进行优化。同时，因为嵌入式领域的应用需求往往多种多样，而一旦部署之后，应用又较为单一，因此可配置处理器目前主要用于嵌入式领域。

一般而言，通常的嵌入式处理器往往也是可配置的，但可配置处理器的独特之处在于：①其可配置的范围更加广阔，不仅仅包括某些关键部件的参数可以调整（如高速缓冲存储器 Cache 参数），或者某些部件可以裁剪，更包括了对于外部接口、指令集的灵活配置的支持；②提供了一整套的可配置设计流程与工具来帮助设计者加速设计流程，比如，其往往提供了一个集成开发环境，设计者可以在环境中模拟分析目标应用（或其关键算法），找出性能瓶颈，在开发环境的帮助下，进行处理器结构调整以及指令调整，并由该环境自动生成针对新结构的编译、模拟等工具，从而使得设计者可以快速地进行新一轮的分析优化设计，并开展设计评估。一旦目标设计确定后，在处理器运行过程中，其结构是不可以调整的（与可重构处理器不同）。

可配置处理器的代表产品：

（1）Tensilica 公司的 Xtensa 处理器是可配置处理器的典型代表。Xtensa 处理器具有不同于其他传统式的嵌入式处理器核心，设计者可以挑选所需的硬件功能模块，并加上自创的新指令与硬件执行单元，就可以设计出针对目标应用定制的处理器内核。

Xtensa 的集成开发环境可以针对每一个定制处理器,自动产生出一套包括操作系统、编译器、模拟器在内的工具链。Xtensa 的指令集拥有专利权,这是一个精简的 16 位与 24 位混合指令集,其基本结构拥有 80 个精简指令集(RISC)指令,其中包括 32 位算术逻辑部件(ALU),6 个管理特殊功能的寄存器、32 个或 64 个普通功能 32 位寄存器,有助于大量缩减编码的长度,从而提高指令的密集度并降低能耗。

(2) Altera 的 Nios Ⅱ 处理器。这是应用于 FPGA 的软核处理器。Nios Ⅱ 系列支持用户增加专用指令,用户能为系统中使用的每个 Nios Ⅱ 处理器创建多达 256 个专用指令,这使得设计者能够细致地调整系统硬件以满足性能目标。专用指令的逻辑和 Nios Ⅱ 自有的指令相同,能够从两个源寄存器取值,可选择将结果写回目标寄存器。Nios Ⅱ 处理器具有完善的软件开发套件,包括编译器、集成开发环境、实时操作系统等。使用 Nios Ⅱ 软件开发工具能够为 Nios Ⅱ 系统自动生成适用于可配置处理器的工具链。

发展趋势

(1) 应用于超级计算领域。超级计算领域的大规模应用的种类实际上不多,很多超级计算机的设计目标通常就是有限的几个大规模并行计算应用。因此,可以借鉴针对应用定制处理器内核的设计思想,对超级计算应用开展处理器定制,以提高系统的性能、降低功耗。

(2) 增强处理器内核的互连功能,以构置较大规模的片上多核处理器。

(3) 提升双精度浮点处理功能。 (张悠慧)

keshi biancheng yuyan

可视编程语言(visual programming language) 一类用图形符号描述计算任务的处理对象和处理过程的语言。这种语言与传统程序语言的最大区别是:传统语言是由正文形式表示的一维字符串结构,而可视语言则是由图形符号的空间排列所表示的多维结构。

现实生活中,图形和图表往往比文字更能形象地表达人们的思想意图,也更有助于思想的了解和交流。由此早在 20 世纪 50 年代计算机科学中就引入了描述计算过程的逻辑图概念,以后又把它逐步规范化为流程图,用它们描述算法,作为进行**程序设计**的辅助工具。60 年代它就成了最早的可视语言,1966 年 W. R. Sutherland 第一个用逻辑图作为可视语言表示程序并给予解释性实现,开创了可视语言的先例。1969 年 T. O. Ellis 等人建立的 Grail 首次用流程图表示计算步骤并直接编译成执行程序。这可能是可视语言的最早编译实现,其中流程图框中内容是用**机器语言**陈述的。以后又建立了很多流程图和流程图变形(如 N-S 图)式的可视语言。非流程图式可视语言的最早成果大约是 C. Christensen 的 AMBIT/G(1968 年)和 AMBIT/L(1971 年),它把**程序**和数据都表示成有向图。程序用模式匹配进行操作,相当复杂的算法(如存储碎片收集)都可以生动地描述成图上的局部变换。第三类可视语言称为 HiGraphs,是 D. Harel 在 1988 年建立的,它允许结点中包含其他结点,它限制产生专用的可视语言,例如 Miro(1988 年)就是一种定义操作系统安全约束的 HiGraphs 语言,StateMaster(1989 年)则是一种计算机用户界面的 HiGraphs 语言。第四类可视语言如 M. Zloof 等人建立的 QBE(1977 年),它允许用户用二维表格指出在关系数据库上的查询。1981 年又把这种思想扩充到办公自动化领域建立 OBE。第五类是 Petri 网式语言。1981 年 T. Ae 等建立的 MOPS-2 用带色 Petri 网按可视方式构造和模拟并发系统。VERDI(1987 年)是一个 Petri 网式可视语言的解释性系统。用户在这种程序执行时可以看到令牌在网上移动的程序动画。第六类是数据流图式可视语言,它们以 PROGRAPH(1983 年)和 Lab-VIEW 为代表。第七类是自动生成用户界面的可视语言,例如 Peridot(1988 年)和 UIMS(1989 年)。此外,还有一些其他类型的可视语言。

这些可视语言都体现了用图形来进行编程的特点。其图形符号大都是软件工作者所熟悉的。近几年对其他(如商业营销、企业管理)领域的对象图形和操作符号也进行了一些讨论,可以建立针对这些领域工作者所熟悉的图形符号的可视语言。

G. TorTora 提出一种可视语言的形式描述:可视语言程序由基本图符的空间排列组成。基本图符分为基本对象图符和对这些对象的处理图符。可视语言的终结符号集包括基本图符和空间操作符两部分。最基本的空间操作符至少包含横向连接、纵向连接和空间覆盖等。一个可视语言表示成三元组(ID,Go,B),其中 ID 是基本图符字典,它刻画了每个基本图符的逻辑部分(含义),物理部分(显示图形)和类型(对象或处理)。Go 是上下文无关文法,它指出如何用空间排列基本图符构造合法的复合图符。每个图符,不论是基本的还是复合的,都具有逻

辑部分和物理部分的对偶表示。一个可视程序也是一个复合图符。B 是知识库,它包含了为构造一个已给可视程序的含义所需的具体领域的信息。(ID,Go,B)构成一个可视属性文法 VAG,它以 Go 为基础。VAG 的语义规则是与 Go 中的产生式相关联的,它指出已给派生树如何合成可视程序的含义。实际的语义规则由包含在 B 中的信息以具体例子说明每个规则的未定部分。

可视语言的实现要求为它建立一个实现环境,称为可视程序开发环境。图符编辑程序负责编辑可视程序 P 的图符,经模式分析程序把空间结构的图形程序变换成模式串,语法分析程序由模式串产生 P 的分析树,再经语义映射程序得到 P 含义(程序的机内表示),再通过解释程序执行或经编译程序产生可执行代码。为了支持大型系统,环境中还应该有数据库、视图工具、浏览程序、项目管理程序等工具。如 1990 年 M. Hirakawa 等人建立的 HiVisual 可视程序环境就支持对程序开发和系统操作的导航,对基于面向对象概念的可视程序和系统操作提供解释机制,提供程序的自顶向下和自底向上开发,提供对现有系统的集成。

可视语言现在仍处于实验研究阶段,要真正走向实用还有相当距离。可视语言以问题空间的处理对象为单位定义基本图符。在某种意义上说它也是面向对象的。而且把对象图像化更易于人们的使用和理解。所以它是很有发展前途的。

参考文献

Ichikawa T,Jungert E,et al. Visual languages and applications. NewYork:Plenum Press,1990 (李万学)

keshi chengxu sheji

可视程序设计(visual programming) 用**可视语言**编写可视程序的方法与过程。

在现实生活中用户所见到的绝大多数对象都是多维的,传统的程序设计要求把这种多维的对象强行变为一维的符号串描述才能被计算机所接受。这就好比计算机是瞎子,只能听用户通过口述把多维对象描述成规定格式的一维文字一样,给用户增加了很大的负担。如果用户能直接把这些多维对象送入计算机,并在计算机中规定了直接对这些多维对象的处理操作,将给用户使用计算机带来很大的方便。现有的一些可视程序设计系统表明,非程序员用户经过短期培训就可以用可视语言编写出相当复杂的程序,这是因为可视语言确实更适合人们的习惯。目前已经建立了大量的可视程序系统,几乎把软件工作者用过的所有辅助编程的图形(如流程图,数据流图,Petri 网图,有向图等)以及这些图形的变种都作为模型建立了各种可视语言。直接用这些基本图形符号,制定一套对这些图符的处理操作,再规定一套语言的**语法**和**语义**规则就形成了一个完整的可视语言。用这种语言编写的可视程序也是一个图形。

可视程序设计不包括用传统正文语言定义图像的系统,如 Macintosh Toolbox 或 X-11 Window Manager Toolkit 等,也不包括绘图软件包的系统,如 Apple Macintosh MacDraw,因为它们并不是用图符来产生程序。

用可视语言编制可视程序应该在一个可视程序设计开发环境下进行。用户首先用它熟悉的图符结合其逻辑含义构思一个解题的图形,如果有图形输入设备,用户可将构图画在纸上用图形扫描器输入,没有图形输入设备,对确定的可视语言建立一个可视程序编译程序,这个编译程序的用户界面中列出了所有基本图符的图形作为选单的内容,用户编写可视程序时可在屏幕上通过鼠标指点选单上的图形拼凑成解题的可视程序以后就等待计算机去执行这个程序。

在执行过程中,根据屏幕提示,输入相应的数据,用户使用起来非常方便。当然,要建立一个这种可视程序开发环境系统的任务是繁重的,除了建立上述编辑程序之外,还要对基本图符给出逻辑含义。在编辑图形的同时或以后,系统根据可视语言的语法和语义导出拼凑的可视程序的逻辑含义(即计算任务和计算步骤),自动解释执行这个可视程序或通过编译程序自动产生该可视程序的执行代码。如果定义的基本图符所表示的对象的图形屏幕形像和逻辑含义都符合用户习惯,用户用可视语言开发自己的软件就相当容易。所以可视语言逐渐向行业(或专业)靠近,变成行业(或专业)专家的语言将成为发展方向。

目前可视程序设计系统基本上仍是一些小型的实验性系统,离实用化的目标尚远,尤其是对复杂图形的(大)可视程序,如何使图形分层,降低复杂程度,达到使其结构清晰、易读等都是需要努力的。

参考文献

1. Chang S-K. Principles of visual programming systems. Prentice Hall Inc. , 1990

2. Chang S-K. Visual language and visual programming. New York: Plenam Press,1990 (李万学)

keshi dianhua

可视电话(video phone)　使通信双方在进行通话时可以互相看到对方图像的一种电话系统。可视电话业务是一种点到点的视频通信业务。可视电话在传输信道上可分为PSTN网(公用电话网)、ISDN网(综合业务数字网络)、IP网(如互联网)和专用网等多种方式。由于传输带宽的限制,可视电话一般使用较小的屏幕和较低的视频帧率。在PSTN网上工作的可视电话,每秒钟可以传输10~15帧画面;在ISDN网上工作的可视电话,每秒钟可以传输15帧以上的画面。1996年国际电信联盟公布了PSTN多媒体可视电话国际标准ITU-H.324标准。

可视电话产品主要有两种类型,一类是以个人电脑为核心的可视电话,除电脑外还配置有摄像机(或小型摄像头)、麦克风和扬声器等输入输出设备;另一类是专用可视电话设备(如一体型可视电话机),它能像普通电话一样,直接接入家用电话线进行可视通话。由于普通电话线普及率很高,因此,随着社会对电话通信多样化需求的增长,电话将从单一的话音通信发展为音视频(及数据)一体的多媒体通信形式。在公用电话网上工作的可视电话最具发展潜力,可视电话作为一种新型的通信方式将进入千家万户。1964年美国贝尔实验室提出了解决方案,直到数字压缩(音视频编解码)、数字通信及数字集成电路等技术的发展和成熟,1992年美国AT&T公司才正式推出2500型可视电话。2002年中国电信推出了“新视通”,中国联通推出了“宝视通”,中国铁通推出了“全视通”可视电话业务。

可视电话主要组成部分有:音视频输入/输出、音视频编解码器、数据处理器、复用/解复用器及网络接口单元等。

可视电话不仅适用于家庭生活,而且还可以广泛应用于各项商务活动、远程教学、保密监控、医院护理、医疗诊断、科学考察等多种不同行业和领域。

(钟玉琢　孙立峰)

keshi fenxixue

可视分析学(visual analytics)　一门由可视交互界面为基础的分析推理科学。它起源于21世纪之初,综合图形学、数据挖掘和人机交互等技术,以可视交互界面为通道,将人的感知和认知能力以可视的方式融入数据处理过程,形成人脑智能和机器智能优势互补和相互提升,建立螺旋式信息交流与知识提炼途径,完成有效的分析推理和决策。

在新的信息时代,人类处理和分析数据的能力已远远落后于获取这些数据的能力。新时期科学发展和工程实践的历史表明,智能数据分析所产生的知识与人类掌握的知识的差异正是导致新的知识发现的根源,而表达、分析与检验这些差异必需人脑智能的参与。为了有效结合人脑智能与机器智能,一个必经途径是以视觉感知为通道,通过可视交互界面,促成人脑和机器智能的双向转换,将人的智能特别是“只可意会,不能言传”的人类知识和个性化经验可视地融入到整个数据分析和推理决策过程中。这个过程构成了一个重要的交叉信息处理思路:可视分析。

传统的可视化是一门利用人眼的感知能力对数据进行交互的可视表达以增强认知的技术。它和智能数据分析的目标都是从数据中获取知识,但手段不同。与可视化相比,可视分析的基本特点是智能的双向转换,即数据分析被嵌入到可视界面中,用户通过可视化和交互方法对机器智能的结果给出反馈和修正,完成人类智能向机器智能的转换,增强后的机器智能将导致新的分析结果,依次循环实现螺旋式的知识提炼。

可视分析学的基本要素包括复杂数据表示与变换、可扩展的数据智能可视化和支持用户分析决策的交互方法与集成环境等。它导引的分析推理模式,是探索复杂数据中蕴含的新规律和新现象的催化剂。如何结合相关学科的方法,研发面向各个应用领域的高效可视分析系统是一个持久的研究话题。

参考文献

1. Thomas J J, Cook K A. Illuminating the path: the research and development agenda for visual analytics. National Visualization and Analysis Ctr, 2005

2. Tufte E. Envisioning information. Graphics Press, 1990

3. Ware C. Information visualization: perception for design. 3rd ed. San Francisco, CA: Morgan Kaufmann, 2012

(陈为)

keshihua

可视化(visualization)　一门利用人眼的感知能力对数据进行可交互的可视表达以增强认知的学科。可视化将数据转化为可感知的图形、符号、颜色、纹理等,来增强数据识别效率,传递有效信息。

人眼是一个高带宽的巨量并行视觉信号输入处理器,具有很强的模式识别能力,对形象化符号的感知速度比对数字或文本快多个数量级,且绝大多数是下意识的。简单地说,可视化的实质是将数据以形象直观的方式展现给用户,让用户以视觉理解的方式获取数据中蕴含的信息。从研究的对象和内涵角度看,可视化可以分为科学可视化、**信息可视化**和**可视分析学**。

1987 年 2 月美国国家科学基金会在华盛顿召开了有关科学可视化的首次会议,会议一致认为:将图形和图像技术应用于科学计算,是一个全新的技术领域,会议将这一技术定名为**科学可视化**。20 世纪 90 年代中期,非结构化、非几何的抽象数据如金融交易、社交网络、文本数据的大量涌现,促生了信息可视化的热潮。进入 21 世纪,现有的可视化技术已难以应对海量、高维、多源和动态数据的分析挑战,需要综合可视化、图形学、数据挖掘理论与方法,研究新的理论模型、新的可视化方法和新的用户交互手段,辅助用户从大尺度、复杂、矛盾甚至不完整的数据中快速挖掘有用的信息以便做出有效决策。这门新兴的学科称为可视分析学。2005 年,美国国家科学基金会联合美国国家卫生研究所召集了一个新的专题小组,讨论可视化研究的现状和面临的挑战,并于 2006 年出版了一个专题报告。与此同时,2004 年美国国土安全部为了应对恐怖袭击,成立了国家可视分析中心,并于 2005 年出版“可视分析的研究和发展规划”,全面阐述了可视分析的挑战。

可视化的主要研究内容包括三个方面:

(1) 数据表示与变换　数据可视化的基础是数据表示和变换。为了允许有效的可视化、分析和记录,输入数据必须从原始状态变换到一种便于计算机处理的结构化的数据表示形式。通常这些结构内蕴于数据本身,需要研究有效的数据提炼或简化方法以最大程度地保持信息和知识的内涵和相应的上下文。构建海量数据的有效表示的主要挑战在于表示方法的可伸缩性和扩展性,以便忠实地保持数据的内容。此外,将不同类型、不同来源的信息合成到一个统一的表示,使得数据分析和处理者能及时聚焦于数据的本质也是研究重点。

(2) 数据的可视显示　欲将数据以一种直观、容易理解和操纵的方式呈现给用户,需要将数据表达转换为可视表示。由于大量数据的采集是以流的形式实时获取,针对静态数据发展起来的可视化显示方法不能直接拓展到动态数据。这不仅要求相邻时间段所产生的可视化结果有一定的连续性,还要求可视化方法达到高效以便实时反馈。因此不仅需要研究新的软件算法,还需要开发更强大的计算平台(如**分布式计算**或**云计算**)和显示平台(如 1 亿像素显示器或大屏幕拼接)。

(3) 用户交互　交互在通过可视化辅助分析决策过程中起重要作用。有关人机交互的探索已经持续很长时间,但智能、适用于海量数据可视化的交互技术,如任务导向的、基于假设的方法还是一个未解难题。其核心挑战是新型的可支持用户分析决策的交互方法。这些交互方法涵盖底层的交互方式与硬件、复杂的交互理念与流程,更需要克服不同类型的显示环境和不同任务带来的可扩充性难点。

从可视化的三个方面看,科学可视化的研究重点是带有空间坐标和几何信息的医学影像数据、三维空间信息测量数据、流体计算模拟数据等。由于当今数据的规模通常超过图形硬件的处理能力,如何快速地呈现数据中包含的几何、拓扑、形状特征和其演化规律是研究的热点。信息可视化的核心问题主要有高维数据的可视化、数据间各种抽象关系的可视化、用户的敏捷交互以及可视化有效性的评断等。可视分析更加关注于机器智能和人脑智能的双向转换,整个探索过程是迭代的、螺旋式上升的过程。迄今为止,可视分析的基本理论与方法仍然是一个有待解决的新课题。

发展到 21 世纪的可视化是一个涉及**数据挖掘**、人机交互、**计算机图形学**、心理学等的交叉学科。如何将这些学科的相关知识有机地结合到一起,开发高度集成的可视分析系统是未来一个重大的研究课题。它在国民经济和国防安全的各个领域都会引申出重大应用难题,如天气预报、数字城市、金融安全、社会网络等。

参考文献

1. Thomas J J, Cook K A (eds.) Illuminating the path: the research and development agenda for visual analytics. (http://nvac.pnl.gov/agenda.stm), 2005

2. Hansen C D, Johnson C R. The visualization handbook. Elsevier, 2005

3. 石教英,蔡文立. 科学计算可视化算法与系统. 北京:科学出版社,1996

(唐泽圣　陈为)

kehu-fuwuqi jisuan

客户-服务器计算(client/server computing)

把复杂任务从功能上分割成几个不同的部分,再

分配到具有前后端结构的**分布式计算环境**中，在前端客户机上运行应用程序，而后端**服务器**则提供某些特定服务的一种计算模式。服务器提供的服务有数据库服务、文件服务和通信服务等。客户-服务器计算环境的基本组成如图 1 所示。客户-服务器计算环境通常由以下几个部分组成：①通信网络，可以是**局域网**或广域网；②服务器，可以是**个人计算机**、**工作站**、**小型计算机**、**大型计算机**等；③客户机，一般是个人计算机；④应用程序开发界面，例如结构查询语言（SQL）、**数据库语言**等；⑤图形用户界面，例如 Windows，Motif 等。

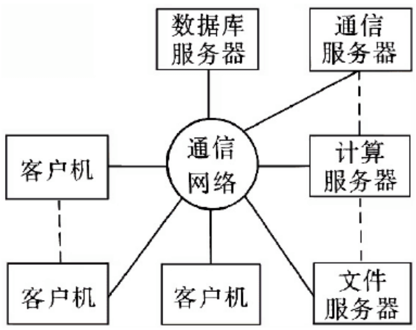


图 1　客户-服务器计算环境的基本构成

客户-服务器计算环境起源于 20 世纪 80 年代中期，它不同于当时的个人计算机的单一式处理方式，也不同于小型计算机与大中型计算机的宿主式计算方式以及计算机网络的点对点计算方式，而是一种能够通过通信来共享位于不同地域的计算机服务器的软、硬件资源的计算模式。

支持客户-服务器计算的主要技术有 3 种。即远程过程调用、分布式数据库管理系统和通信协议。

（1）远程过程调用　指用户可以像调用本地过程一样调用不同地域的不同计算机上的过程，从而使得应用程序设计人员不必设计和开发有关发送和接受信息的实现细节。实现远程过程调用必须解决好下述 4 个问题：①存根子程序设计。存根子程序帮助用户掩盖远程过程调用的细节，使用户像调用本地过程那样调用远程过程。②绑定问题。绑定程序使用户程序和特定的服务程序发生联系，也就是正确定位用户程序所需要的服务程序，从而使远程过程调用成为可能。③可靠性问题。④参数传递问题。即在进行远程过程调用时，如何在不同机器之间正确有效地传递参数。

（2）分布式数据库管理系统　将分散在不同结点上的数据资源通过通信网络连接起来，统一进行管理和控制，使其在逻辑上形成一个统一的数据库系统（参见**分布式数据库系统**）。

（3）通信协议　为远程过程调用和分布式数据库管理系统提供通信规则，从而使得客户-服务器计算成为可能（参见**网络协议**）。

客户-服务器计算环境具有如下优点：

（1）运行性能高　在分散处理的应用环境中，由于客户-服务器计算把处理功能进行了适当的划分，显著地减少了网络上的流通负担，提高了整个系统的运行性能。由于服务器端不需支持应用功能，使服务器端的内存和中央处理器可全部供数据库管理系统等服务器中的后台程序使用。客户机端主要执行应用程序，不需提供数据管理功能，因此，客户机端的内存和中央处理器可全部提供给应用程序使用。另外，数据库服务器提供多种不同层次的锁定和快速复制功能，从而提供了更好的并行控制以确保多个用户在同一时间内存取相同的数据，这进一步提高了运行性能。

（2）便于最终用户使用　在客户-服务器计算环境中，客户机可根据最终用户需要装载文档处理软件、决策支持工具、前端电子邮件、数据库请求程序以及图形用户界面等，从而客户机为最终用户提供比宿主计算环境更为方便的用户界面。

（3）具有多功能的应用程序开发接口　客户-服务器计算环境提供由客户机用户使用的标准语言（例如 SQL）和多种开发支持工具，这些标准语言和支持工具构成方便的多功能应用程序开发接口。用户可以使用这些应用程序开发接口透明地请求服务器的各种有效服务。

（4）开放的体系结构　客户-服务器计算环境具有开放的体系结构，它们大都使用国际流行协议 TCP/IP 或 SPX/IPX，从而使得支持这些协议的不同厂家的软、硬件产品可以集中到同一客户-服务器计算环境中。

（5）良好的可扩充性　由于客户-服务器计算环境具有开放的体系结构以及应用程序和服务程序分别在客户机和服务器上执行，客户-服务器计算环境具有良好的可扩充性。可以在后端服务器功能不变的前提下，对服务器硬件或软件进行扩充或更新换代，这种更新换代不会给前端应用程序带来负面影响。同样，在后端服务器负载能力允许的条件下，可以自由增加客户机的台数。客户-服务器计算环境的可扩充性不仅使用户在应用软件方面的投资得到保障，并为用户提供一种低消耗的逐步更换设备

的途径,从而使得用该技术构筑的信息系统能方便迅速地跟上新技术的变化。

建立客户–服务器计算环境的主要难点在于:

(1) 安全性问题。客户–服务器计算环境将应用程序和服务信息分布在整个计算环境中,且允许用户访问外界信息或允许外界用户访问计算环境中的信息,因此,必须设置相应的访问权限、口令、密码或其他保密措施。

(2) 故障隔离问题。

(3) 数据的完整性和一致性问题。

(4) 正确划分应用程序,以获得最佳系统性能问题。

(5) 应用程序开发人员、使用和管理人员的教育与培训问题。

(6) 系统维护问题。

客户–服务器计算模式已日益普及,随着网络技术的成熟,越来越多的用户将采用这一模式设置计算机系统,但有些用户从降低维修成本、保证系统安全性等角度出发,仍会采用宿主–终端计算模式。

(张尧学)

kehu-fuwuqi moshi

客户-服务器模式(client/server mode) 参见**客户-服务器计算**。

kehu fuwuqi moxing

客户/服务器模型(client/server model) 参见**客户-服务器计算**。

kongfen fuyong

空分复用(space division multiplexing) 空分复用技术利用对空间的分割来达到共享资源的目的。

空分复用的基本原理是:不同用户的地理位置不同,采用天线的波束成形技术,使不同的波束方向对准不同的用户,这样,不同的用户即可共享频率资源、时间资源和码资源。如图1所示。

通过空分复用,可以在固定频段内提高通信的容量,该方法通常用于移动通信系统中。

参考文献

1. Tanenbaum A S, Wetherall D J. 计算机网络. 北京:清华大学出版社, 2012

2. 纪越峰. 现代通信技术. 北京:北京邮电大学出版社, 2010

(袁华)

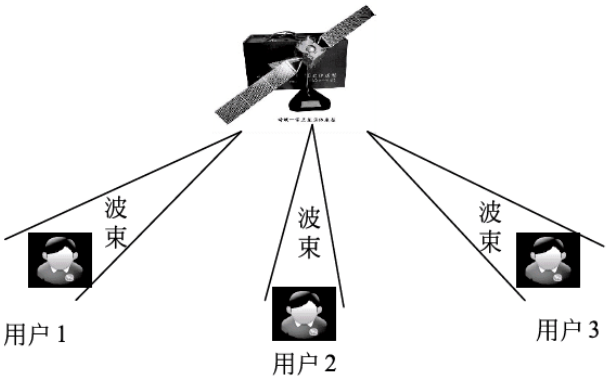


图1　空分复用技术原理示意图

kongjian fuzaxing

空间复杂性(space complexity) 解答问题时以空间度量的算法复杂性。算法的空间需求是指计算所需要的计算机存储量。在图灵机计算模型中,使用的空间量定义为读写头访问的不同的带方格数。因为访问的带方格数不可能多于计算步骤,所以任何在 $T(n)$ 时间内可解的问题也必可在 $T(n)$ 空间内可解。更精确地说,输入本身占有的空间是不应该计算在空间用量内的,所以图灵机的带分成输入带和工作带,空间复杂性是指图灵机运行时带头读写工作带上经过的带方格数。如果是随机存储器模型,空间复杂性是指工作单元数(有时按工作单元的二进位数)。大量实例表明,空间复杂性与时间复杂性一样,也是一个与计算模型无关的概念。

在多项式时间内可解的所有问题都可以在多项式空间内解决,但是否存在在多项式空间内可解的问题不能在多项式时间内解决,这仍然是一个未解决的问题。一般认为有这样的问题存在。此即著名的 $P \neq PSPACE$ 问题。

用图灵机计算模型,可以得到很简单的带压缩定理:如果语言 L 能够被图灵机 M 在空间 $s(n)$ 内接受,c 是任意(0,1)之间的实数,则 L 可以被另一图灵机 M' 在空间 $cs(n)$ 内接受。

这表明在空间复杂性函数中,常系数是一个次要的因素。人们更为关心的是空间复杂性在输入规模趋于 ∞ 时的渐近性态,即它的增长率。

还有一个与时间复杂性截然不同的定理是:如果语言 L 能够被不确定图灵机 M 在空间 $s(n)$ 内接受,则 L 可以被另一确定图灵机 M' 在空间 $s^2(n)$ 内接受。

并行算法的空间复杂性和时间复杂性有相互折算的关系,空间耗费大,即并行程度高,则并行算法

的执行时间可以降低,但也不是说空间用量任意地增加,并行时间复杂性可以任意地小,故权衡并行算法的时空折中关系,设计好的空间体系结构(如 *n*-维立方体、星形图等)是并行算法空间复杂性研究主要关心的问题。

参考文献

1. Hopcroft J E, Ullman. 自动机理论、语言和计算. 徐美瑞,译. 北京:科学出版社,1986

2. Balcázar J L, Díaz J, Gabarró J. Structural complexity, Ⅰ,Ⅱ. Springer-Verlag, 1988

3. Akl S G. The design and analysis of parallel algorithms. Englewood Cliffs, NJ: Prentice Hall

(朱洪)

kongjian luoji

空间逻辑(spatial logic) 一种描述空间几何实体的结构特征及其相互关系的逻辑语言。空间逻辑使用的语法可以是**一阶逻辑**、一阶逻辑的片段或**高阶逻辑**,所涉及的空间结构可以是任何一种几何空间,如拓扑空间、仿射空间、度量空间、投影平面或三维欧氏空间等。空间逻辑目前主要包括拓扑空间逻辑、度量空间逻辑、方向关系逻辑、拓扑与度量结合的空间逻辑等,其研究热点主要集中在逻辑公式的有效性、表达能力、计算复杂性和解释等四个方面。

关于空间逻辑的最早工作可能是 1899 年 David Hilbert 在其著作"The Foundations of Geometry"中为欧氏几何提出的 20 条(最早为 21 条)假设(后被称为 Hilbert 公理)。Hilbert 公理系统中包含点、线、面三个基本对象,以及六个基本关系:一个三元关系 *Betweenness*、三个二元关系 *Containment*(分别对应点线面)和两个二元关系 *Congruence*(分别对应线段和角)。20 世纪 40 年代前后 Tarski 和 McKinsey 给出**模态逻辑**的拓扑解释,他们将模态逻辑 S4 中的可能性算子解释成实数空间或度量空间上的拓扑闭包。1959 年,Tarski 等人基于一阶逻辑定义了关于欧氏几何的公理系统(后称为 Tarski 公理)。在 Tarski 公理系统中,基本对象(primitive objects)为点,基本空间关系有两种:三元关系 betweenness 和四元关系 Congruence。与 Hilbert 公理系统相比,该语言足以表示欧氏几何,且可判定。1981 年,Bowman L. Clarke 基于 Tarski 的工作,提出了个体连接演算的三条公理。1992 年,Randell 等人基于 Clarke 的工作提出了描述空间拓扑关系的 RCC 一阶拓扑逻辑,定义了八个基本谓词以表示空间区域间的基本拓扑关系,这也是一阶拓扑逻辑中最具代表性工作。此后研究者从表达能力和计算复杂性对 RCC 进行了完善和扩展。1996 年,Bennett 给出了 S4 在拓扑空间的解释,并增加了两个量词得到空间模态逻辑 S4u,分析了 S4u 的计算复杂性。距离逻辑的代表性工作是 2003 年 Kutz、Wolter 和 Sturm 等人提出的两类表示距离的逻辑语言:一阶度量逻辑(first-order metric logic)FM[M]和模态度量逻辑(Modal Metric Logic)MS[M],其中 M 为任意度量空间。方向关系逻辑主要包括以区域为基元的空间命题邻域逻辑(spatial propositional neighborhood logic)和以点为基元的锥形主方向关系的空间模态逻辑。

现实世界中需要同时表示对象的时间和空间属性,随着对空间逻辑和**时态逻辑**研究的不断深入,同时表示对象的时间和空间属性及其关系已成为可能。广义上讲,描述时间、空间属性及其相互关系的逻辑形式化方法都属于时空逻辑的研究范畴。

时空逻辑的研究最初主要是一阶时空逻辑,而后时空模态逻辑逐渐成为研究热点。1998 年,Muller 按时空统一形式化的思路,从语法出发扩展空间逻辑公理集,以空间区域为基本实体,建立了一阶时空逻辑模型。这是时空逻辑最早的研究工作。2000 年,Wolter 等提出从语义出发构造时空逻辑的思想,此工作对时空逻辑研究产生了较大影响。之后人们开展了 NP-完全的时空模态逻辑 PST,命题时态逻辑 PTL 和 S4u 相结合的时空模态逻辑 PSTL,时空结合逻辑可判定性与复杂性,三模态动态拓扑逻辑 DTL,三维时空相关逻辑等研究。但目前的大多数研究工作都是构建时空混合逻辑,统一表达时间和空间逻辑的研究尚不多见。

参考文献

Kontchakov R, Kurucz A, Wolter F, Zakharyaschev M. Handbook of spatial logic. Springer, 2007

(谢琦 欧阳继红 刘大有)

kongjian shujuku

空间数据库(spatial database) 描述、存储和查询处理空间数据的**数据库**。空间数据库在传统数据类型处理能力的基础上增加了对空间数据的处理能力。它从数据模型、查询语言、查询处理和存储方法诸方面对空间数据提供了全面的支持,是当前**地理信息系统**(GIS)和**计算机辅助设计**(CAD)等应用的基础。

*空间数据*是用于表示空间物体的位置、形状、大

小和分布特征等方面信息的数据,用于描述二维、三维和多维分布的关于区域的现象。空间数据不仅包括物体本身的空间位置及状态信息,还包括表示物体的空间关系(即拓扑关系)的信息。属性数据为非空间数据,用于描述空间物体的性质,对空间物体进行语义定义。

空间数据库除了要提供对空间数据的存储与访问功能外还要提供对常规数据的访问能力,所以大多数空间数据库是以现有成熟的**数据库管理系统**为基础建立的。其系统结构如图 1 所示。

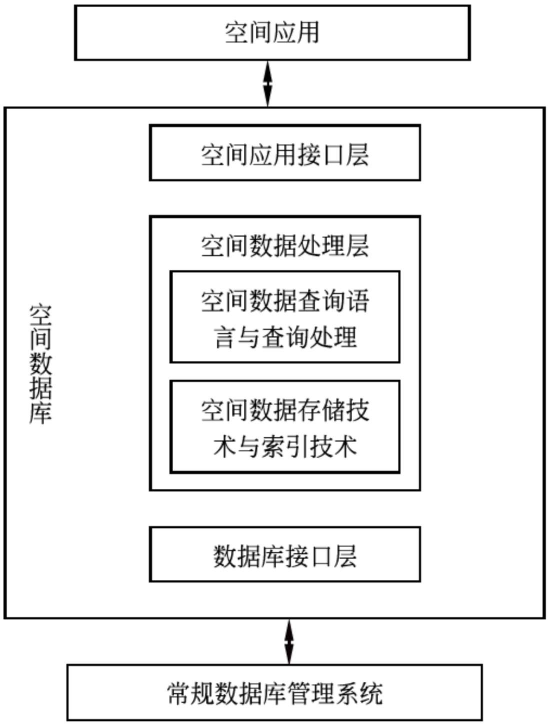


图 1　空间数据库系统结构图

上层是各种空间应用,如 GIS 应用、CAD 应用等;中间层是空间数据库系统,它结合传统的数据库技术实现对空间对象的存储与查询,并提供对空间应用开发的支持;下层是常规的数据库管理系统,一般是**对象-关系数据库**管理系统和**面向对象数据库**管理系统,实现对常规数据的存储和查询。

空间数据库部分也分为 3 层,其中空间应用接口层提供应用的访问接口,包括抽象数据类型、数据模型、可视化接口等。空间数据处理层负责实现对空间数据的存储管理和查询访问。查询语言多是以结构查询语言(SQL)为基础,增加相应的函数实现对空间对象和空间关系的查询。查询操作主要由空间选择操作和空间连接操作构成。为了提高访问效率,引入了针对空间数据的索引结构。常用的索引结构可分为面向空间点的索引结构(如网格文件、K 维树、自适应 K 维树等)和面向矩形的索引结构(如 R 树、四叉树和单元树等)。空间选择操作用于查找满足某个空间约束的空间对象对的集合。数据库接口层负责与底层数据库系统之间的融合,处理常规数据的查询,并与空间数据处理层一同实现对空间对象的查询。

近年来,随着手机设备的普及,基于位置的服务成为新的研究热点。因此面向道路网络的空间对象管理和移动对象管理等方面的技术成为空间数据管理中的新内容。在基于道路网络的空间对象管理技术中,空间数据管理中常用的欧式距离由道路网络上的距离代替。移动对象管理技术实现了对大量不断移动对象的空间位置信息的实时存储管理和查询以及对对象的移动轨迹的管理和分析。

参考文献

1. Guting R H, Fernuniverstat H. Spatial database system VLDB'94, 1994

2. Shekhar S, Chawla S, et al. Spatial databases: accomplishments and research needs. IEEE Trans. on Knowledge and Data Engineering, 1999,11(1)

(汪卫　施伯乐)

kongzhigan

控制杆(joystick)　一种用手操纵来直接控制屏幕上光标移动的杆状**输入设备**。它的外形是一个细杆,装在方形的底座上。杆长 2.5 ~ 10 cm,杆上有手把,用手抓杆可以前后左右晃动。底座内有由电位器、开关、光学编码盘等构成的传感器,用来检测杆的运动方向、位移和速度。当使用者操纵此杆偏离其中心位置时,屏幕上的光标也作相应方向和位移的运动。手把上有按钮,用来执行某种操作。

控制杆大量应用于游戏机,这时它是一个相对定位的设备。控制杆也用于工业控制和军事方面,这时有的控制杆带有恢复弹簧,当不受力时会回到中心位置,因此也可以作为绝对定位设备。

控制杆精度低,但操作方便,价廉。

在个人计算机中,控制杆用电缆与主机连接,在主机板上有专用的端口。但随着半导体芯片集成度的提高,已不需要用整个插件做游戏端口了,此端口通常和异步通信电路等做在一块卡上。　(林兼)

kongzhiqi

控制器(control unit)　计算机的控制中心和指

挥中心。它负责控制计算机各部件运行程序，执行指令，完成程序规定的功能。程序是一个指令序列，控制器需要按照程序的要求，决定指令执行顺序，取出当前应该执行的指令，生成各种操作控制命令，逐条完成各条指令的功能。

控制器的基本功能

控制器的基本功能有 4 项：

(1) 控制各条指令执行的顺序。

(2) 生成实现指令要求的各种操作控制命令，控制有关部件完成指令规定的功能。

(3) 处理运行程序过程中的异常情况，如运算结果溢出、数据校验出错以及输入输出中断请求等随机操作。

(4) 向全机提供统一的时序控制信号，协调各部件操作。

控制器的组成

控制器的逻辑框图如图 1 所示。

(1) 指令部件　按照当前要执行的指令地址，取出指令，保存指令，分析指令。指令部件包括指令计数器，又称程序计数器 PC，存放本条指令的主存地址，本指令结束时，自动生成下条指令地址。指令寄存器 IR，存放正在执行的指令。指令译码器 ID，用于解释二进制的指令操作码的功能，给出控制电位。

(2) 地址部件　由地址加法器，变址寄存器 R_x、基地址寄存器 R_b 等组成。根据指令格式中寻址方式的规定生成被运算数据在主存中存放的有效地址。如变址寻址时，操作数有效地址 $E_a=(R_x)+D$，其中 D 为指令中的位移量，有时叫形式地址。基址寻址时，操作数有效地址 $E_a=(R_b)+D$。相对寻址时，$E_a=(PC)+D$ 等。

转移指令也可根据寻址方式、转移条件生成下条指令的有效地址，指令结束时送入指令计数器 PC 中。在顺序执行的程序中，下条指令地址由 (PC) + 1 得到。

(3) 时序部件　也叫时序系统，提供机器工作的时间顺序控制信号。计算机执行一条指令的时间叫指令周期，完成一条指令又分为若干阶段，如取指令、取数、执行操作码规定的操作等，这些时间阶段称为**机器周期**，如取指周期、取数周期、执行周期、中断周期等。一个机器周期内的操作又可分为几步完成，其先后顺序由节拍电位来控制。一个完整的时序系统可分为指令周期、机器周期、节拍电位、工作脉冲等不同层次，分别对应指令执行过程中不同阶段的不同操作。现在机器中，常把机器周期与节拍电位合并为时钟周期。

(4) 操作控制部件　用来生成各种指令、各机器周期、各时间节拍需要的操作控制命令，控制有关部件对应的操作。

(5) 中断逻辑　在机器运行程序过程中，出现运算错(如溢出)、硬件错(如掉电)以及输入输出中断请求时，需要停止当前运行的程序，转去为紧急事件服务，服务完毕又回到原来程序继续执行，这个过程由中断逻辑完成。

控制器分类

控制器按照控制方式可分为同步控制与异步控

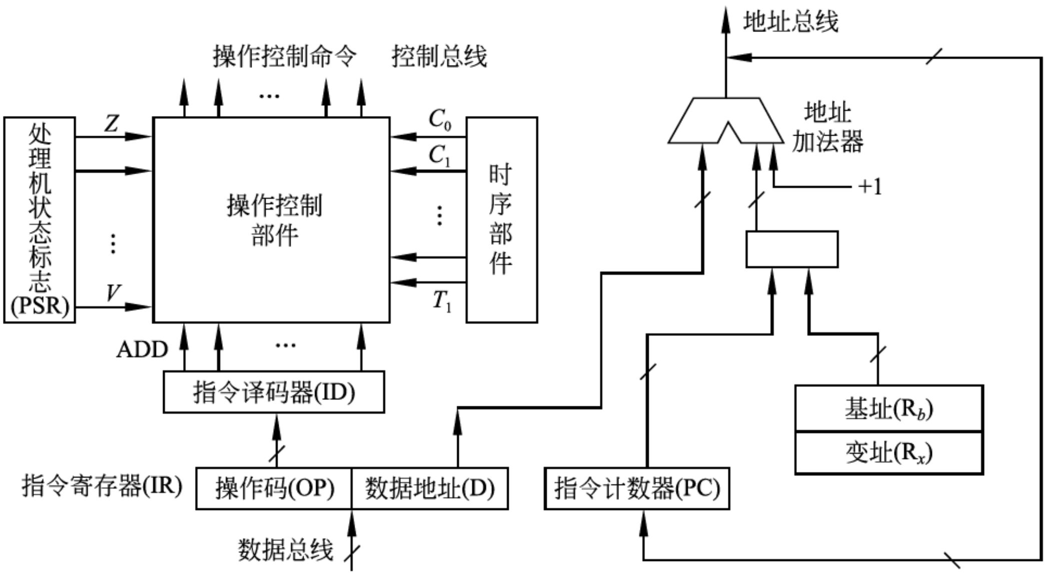


图 1　控制器逻辑框图

制。按照生成操作控制命令的实现方案又可分为组合逻辑控制器(参见**硬连线控制器**)和微程序控制器(参见**微程序控制器**)。

为了减少中央控制部件的复杂性和负担,常常把控制逻辑分布到各功能部件上,因而又可分为集中控制与分布控制。

并行处理技术、流水线技术和超级计算机的出现,大大提高了计算机的运算速度,也使控制器的结构更为复杂。

参考文献

1. 金兰,金波. 计算机组织:原理、分析与设计. 北京:清华大学出版社,2006

2. 谢树煜. 计算机组成原理. 2 版. 北京:清华大学出版社,2009

(谢树煜)

kuaikeca biancheng zhidu cunchuqi xinpian

快可擦编程只读存储器芯片(flash erasable programmable read only memory chip) 一种基于浮栅单管存储单元结构的电可修改的半导体只读存储器芯片,简称快闪存储器或闪存芯片(Flash EPROM 芯片)。闪存芯片在功能上也是**电可擦编程只读存储器(EEPROM)芯片**,能在电路系统中擦除和写入数据(编程)。它使用与**可擦编程只读存储器(EPROM)芯片**相近的单管存储单元工艺,但浮栅与沟道间绝缘层厚度小于 10 nm,用隧道效应擦除浮栅上的电荷,因而具有单管的高集成度和可采用塑料封装等低成本的特点。由于可实现扇区(sector)或存储块(block)/全芯片(bulk)擦除,因而称为闪存,以区别于可按一个字节编程的常规 EEPROM 芯片。

闪存芯片根据接口和内部组织的不同,有 NOR、NAND、DINOR 和 AND 四种。应用最广泛的闪存是 NOR 和 NAND 快闪存储器芯片。NOR 闪存芯片内部同一列线线上存储单元的 MOS 管以并联方式相连(类似 NOR 门);而 NAND 闪存芯片内部同一列线线上存储单元的 MOS 管以串联方式相连(类似 NAND 门)。NOR 闪存芯片读出速度快,但存储单元面积大,不适合大容量集成。主要用于数字系统的开机导引程序和 BIOS 的存储和中等容量的数据存储,在个人电脑、手持通信和消费娱乐电子设备等领域广泛使用。而 NAND 闪存芯片读出速度慢,但存储单元面积小,适合大容量集成。NAND 闪存芯片主要用于大容量的数据存储,作移动存储器件或设备。例如 USB 存储卡(U 盘)、音乐/相片/影像的存储卡和固态硬盘就是由 NAND 闪存芯片加接口转换控制芯片等电路封装而成。

由于使用电荷泵技术在芯片内升压,闪存芯片通常用单工作电源。现电源电压已从 5 V 降至 3/3.3 V。在手持移动设备中为降低功耗也有 1.8 V 的产品;或数据输出电路用 1.8 V,内部核心电路仍为 3.3 V。为减少擦除和编程时间,有的 NOR 闪存芯片产品仍保留 12 V 编程电压输入脚,在脱机编程时使用。

NOR 闪存芯片内部存储单元仍按行、列和数据位数组成矩阵,其接口和结构无统一标准。各生产厂根据市场的发展和需求,有各种不同的产品。相互间不一定兼容。

NAND 闪存芯片的接口有统一标准。但从读写控制的角度看,对外以页面/存储块/存储体方式构成。一个芯片可有一个或多个存储体。每个体由多个存储块组成。每个块由多个页面组成。每页的字节数与数据 I/O 的位数(8 或 16)无关。一行可有二个以上页面。一个存储块可有多个行组成。为提高数据读出的可靠性,每 512 B 增加 16 B 冗余,存储出错校正码,根据厂家设计,可校正 1/2/4 位数据错。

表 1 为大/小块二种 NAND 闪存芯片典型芯片主要参数,以作参考。

表 1　NAND 闪存芯片典型芯片主要参数

	页面/块大小	芯片总块数	由存储单元到寄存器时间 t_R	页面读/写数据周期	页编程时间 typ/max	块擦除时间 typ/max
32 Mb × 16	256 W/16 KB	2048	10 μs max	42/40 ns min	200/500 μs	2/3 ms
512 Mb × 8	4 KB/2 MB	2048	60 μs max	50 ns min	410/530 μs	4.5/16 ms

NAND 闪存由于存储容量大,为提高芯片的产出率,并且出厂芯片允许有坏的存储块(block)存在。允许的最多坏块数,一般为总块数的 1% 左右。在坏块的冗余块的固定位置存储非 FFh(对 ×8 芯片),或非 FFFFh(对 ×16 芯片)数据标识。以便在加电后对每个存储块的冗余块固定位置的数据读出

检查,建立或修改坏块地址表。在擦除或编程过程最后验证时发现新的一页面数据有不能校正的错误时,需用未使用的好块来替代有错误页面的坏块。并对此坏块设置坏块标识,并修改坏块地址表。

为适应手持移动设备减小芯片封装面积的需求,厂家提供 NAND 闪存芯片与其他功能的芯片,如 SDRAM(或 DDR SDRAM)、NOR 闪存、准静态 SRAM(NtSRAM)、逻辑控制电路,封装在一起的多芯片封装(MCP)的与特定处理器配套的专用芯片,使原来用多个不同功能封装的多个芯片变为一个封装的多功能芯片,从而减少所占用的印制板面积。

随着半导体工艺的发展,NAND 闪存的容量也在不断地增加。同时为减小页面读的周期,也开始采用双倍数据速率的技术。常规的 NAND 闪存读周期最快 25 ns,即 40 MT/s(MegaTransfer/s)。而 2010 年 8 月宣布的 32 nm 工艺 MLC 单元 64/128/256 Gb DDR NAND 闪存的页面读出周期可达 6.7 ns,传送速率达 133 MT/s。作为 NAND 闪存的衍生产品高速 32 GB SDHC 卡、高速 64 GB SDXC 卡、64 GB USB 3.0 卡的容量已问世,而 SATA 固态盘容量也达到 256/512 GB,读出/写入速度也相应提高。

参考文献

1. Sharma A K. 先进半导体存储器——结构、设计与应用. 曾莹,等译. 北京:电子工业出版社,2005

2. 桑野雅彦. 存储器 IC 的应用技巧. 王庆,译. 北京:科学出版社,2006

3. http://www.samsung.com/Products/Semiconductor

(孙祖希)

kuaishan cunchuqi

快闪存储器(flash memory) 一种非易失性允许多次擦除和写入的**半导体存储器**,简称闪存。传统的电可擦编程只读存储器(EEPROM)只能擦除和重写单个存储位置,而快闪存储器以数据块为基本单位进行擦除,因而具有高速写入的特性。快闪存储器还具有低功耗、抗恶劣环境等优点,被广泛应用于嵌入式系统、消费类电子以及移动设备上,如手机、笔记本电脑、MP3、U 盘、数字相机(俗称数码相机)、游戏机等。

基本原理

典型的闪存结构如图 1 所示。在高电压条件下通过 F-N 隧道效应将电子注入浮动栅门完成数据的写入,根据浮动栅上电荷的有无决定读出数据的

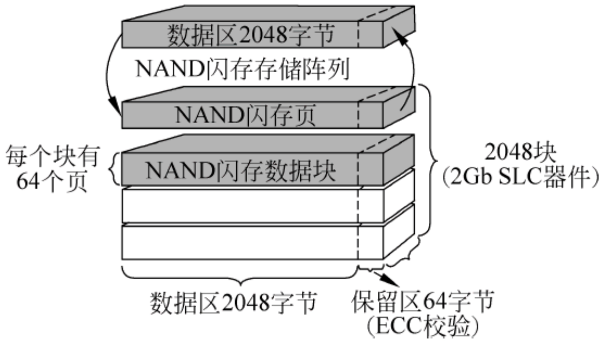


图 1 闪存数据组织结构示意图

"0"和"1"状态。典型基于闪存的存储驱动器读取或写入数据的时间为 0.2 ms,工作状态下最高读取速度为 10 万 I/O PS,最高写入速度为 2.5 万 I/O PS。闪存分为单层存储单元(single level cell, SLC)和多层存储单元(multi level cell, MLC)两种。SLC 指每个单元(cell)只能存储一个数据位(bit),MLC 指每个数据单元能存储两个数据位,所以 MLC 的存储容量是 SLC 的两倍。以 2 GB 的 SLC NAND 型闪存为例,闪存由 2048 个块(block)组成,每个块包含 64 个页(page),每个数据页由 2112 字节组成,其中数据字节为 2048,其余的 64 个字节为校验字节。读、写、擦除速率通常为 25 μs、220 μs、500 μs。由于不同的操作在速度上的差异,数据的读写以页为单位,擦除以块为单位。NAND 闪存中的存储单元不支持原位写入覆盖操作,因此每个存储单元必须先擦除之后才能按序写入。由于介质特性, SLC 的复写次数为 10 万次,而 MLC 的复写次数仅有 1 万次。因此闪存介质面临的最大问题是写容忍度。

发展历史

1984 年日本舛冈富士雄博士发明了闪存,并在加州旧金山 IEEE 国际电子器件大会(International Electron Devices Meeting, IEDM)上发表了这项发明。1988 年 Intel 公司推出了第一款商业性的 NOR Flash 芯片。NOR Flash 因可提供完整的寻址与数据总线,适合取代早期的 ROM 芯片,但其擦除操作需花费的时间很长。1989 年东芝公司在国际固态电路学会上发布了 NAND Flash 成果。与 NOR Flash 相比,NAND Flash 由于其较高的存储密度和较低的成本,较短的擦除时间以及较多的擦除次数而备受关注。NAND Flash 非常适合用于存储卡之类的大量存储设备。第一款创建在 NAND Flash 基础上的可移除式存储媒体是智能媒体卡,此后许多存储媒体也跟着采用 NAND Flash,包括多媒体卡、SD 卡和

存储棒等。

应用

闪存因具有较快的读取速度，而写入操作比较复杂，花费时间长等特点，非常适合应用于各种移动设备和对读取速度要求高的场所，如移动电话、掌上电脑、数码相机、GPS等设备上以及作为大型数据中心、云存储的缓存等。基于快闪存储器的U盘已经成为计算机系统之间传输数据的流行手段，基于快闪存储器的固态盘也逐渐成为主流的存储设备。

参考文献

Grupp L M et al. Characterizing flash memory: Anomalies, observations, and applications. Proc 42nd Ann IEEE/ACM Int'l Symp Microarchitecture (Micro 09). ACM Press, 2009, 24-33 （吴非）

kuaisu Fuliye bianhuan

快速傅里叶变换（fast Fourier transform, FFT） 计算离散傅里叶变换（DFT）的一种快速算法。

一维DFT是把N点序列$x(n)$（$n=0,1,\cdots,N-1$）按线性关系

$$X(k)=\sum_{n=0}^{N-1}x(n)W^{nk},\quad k=0,1,\cdots,N-1 \tag{1}$$

变为N点序列$X(k)$（$k=0,1,\cdots,N-1$）的一种线性变换，其中$W=e^{-i2\pi/N}$，$i=\sqrt{-1}$，它具有如下性质：

$$\frac{1}{N}\sum_{n=0}^{N-1}W^{nt}=\begin{cases}1, & \text{当 } t \text{ 是 } N \text{ 的倍数时}\\ 0, & \text{其他}\end{cases}$$

利用这个性质，可知逆变换（IDFT）是

$$x(n)=\frac{1}{N}\sum_{k=0}^{N-1}X(k)W^{-nk},\quad n=0,1,\cdots,N-1 \tag{2}$$

直接计算N点DFT和IDFT各需要N^2次乘法和$N(N-1)$次加法，当N很大时，运算量非常大。1965年美国J. W. Cooley和J. W. Tukey提出的FFT算法使运算量降为$N\log_2N$，计算效率提高$N/\log_2N$倍，引起了各国学者的广泛注意。50多年来，各种快速算法相继出现，成为数值代数中最活跃的一个领域。

FFT算法的基本思想是利用基函数W^{nk}的周期性和对称性，改变式(1)的计算次序和求和次序，利用递推步骤以减少运算量。例如，当$N=2^t$时，改变式(1)的求和次序，使偶数项和奇数项分别求和，并对$X(k)$的前、后$N/2$个样本分别计算，就可得到

$$X(k)=G(k)+W^kH(k)$$
$$X(k+N/2)=G(k)-W^kH(k)$$
$$k=0,1,\cdots,N/2-1 \tag{3}$$

其中

$$G(k)=\sum_{n=0}^{N/2-1}x(2n)W^{2nk}$$
$$H(k)=\sum_{n=0}^{N/2-1}x(2n+1)W^{2nk}$$
$$k=0,1,\cdots,N/2-1 \tag{4}$$

由此可知，只要先计算式(4)中的两个$N/2$点序列$G(k)$和$H(k)$，然后由式(3)用$(N/2-1)$个乘法和N个加法便可计算出$X(k)$，而$G(k)$和$H(k)$是$N/2$点DFT，可用类似方法计算，如此递推计算下去，$(\log_2N-1)$步后就可变成二点DFT。如果用$M(N)$和$A(N)$分别表示N点DFT所需乘法和加法数，按上述算法有

$$M(N)=\frac{1}{2}N\log_2N-N+1 \tag{5}$$

$$A(N)=N\log_2N \tag{6}$$

其中利用了$M(2)=0$，$A(2)=2$。通常把这种方法叫做时间抽取基-2 FFT算法。还有频率抽取基-2 FFT算法，这只要将式(1)的$X(k)$按下标k为奇数、偶数分别计算，将和式分做前后各$N/2$项相加，就有

$$X(2k)=\sum_{n=0}^{N/2-1}g(n)W^{2nk}$$
$$X(2k+1)=\sum_{n=0}^{N/2-1}h(n)W^{2nk}$$
$$k=0,1,\cdots,N/2-1 \tag{7}$$

其中

$$g(n)=x(n)+x(n+N/2)$$
$$h(n)=[x(n)-x(n+N/2)]W^n$$
$$n=0,1,\cdots,N/2-1 \tag{8}$$

即先用$(N/2-1)$个乘法和N个加法计算式(8)中的$g(n)$和$h(n)$，再计算式(7)中的两个$N/2$点DFT，可用相同办法计算，如此递推计算下去，$(\log_2N-1)$即得结果，运算量也如式(5)和式(6)所示。

由基-2 FFT算法容易推出任意基-r FFT算法。设$N=r^t$（r为任意正整数），按时间抽取法，式(1)可化作

$$X(k+sN/r)=\sum_{l=0}^{r-1}G_l(k)W^{lsN/r}$$

$$s=0,1,\cdots,r-1;k=0,1,\cdots,N/r-1$$

其中

$$G_l(k) = H_l(k)W^{lk}$$

$$H_l(k) = \sum_{n=0}^{N/r-1} x(rn+l)W^{rnk}$$

$$l = 0,1,\cdots,r-1;k = 0,1,\cdots,N/r-1$$

也就是把 $N = r^t$ 点 DFT 简化为 N/r 个 r 点 DFT 和 r 个 N/r 点 DFT 来计算，附加运算量仅为 $(r-1)(N/r-1)$ 次乘法，如此递推 $(\log_r N - 1)$ 步就成为 r 点 DFT，所以基-r FFT 运算量是

$$M(N) = \frac{1}{r}N\log_r N[M(r) + r - 1] - N + 1$$

$$A(N) = \frac{1}{r}NA(r)\log_r N$$

其中 $M(r)$ 和 $A(r)$ 是 r 点 DFT 的乘法和加法量。

当 N 不是单一整数的乘幂，而可分解为 $N = N_1 \cdot N_2 \cdot \cdots \cdot N_r$ 时，又可类似地建立运算量不超过 $N(N_1 + N_2 + \cdots + N_r)$ 的 FFT 算法。例如 $N = 75 = 3 \times 25$，可将 75 点 DFT 简化为 3 个 25 点 DFT 和 25 个 3 点 DFT，而 25 点 DFT 可简化为 5 个 5 点 DFT，因此只要知道 3 点和 5 点 DFT 的快速算法，就可得到 75 点 DFT 的快速算法，这种方法称为混合基 FFT 算法。

上述算法可应用于 IDFT。FFT 算法在数值计算中的一个重要应用是计算**循环卷积**。此外还可用于快速计算多项式乘积、大整数乘积、某些矩阵的逆和特征值等，FFT 在函数逼近论中和在众多实际技术领域中也已有广泛的应用，在光谱、声谱、地震谱分析、晶体结构分析、滤波、数字信号处理、图像信号处理、物探、雷达、卫星摄像分析、全息图，以及心电图、脑电图、X 光相片强化等方面都有大量应用，很有发展前途。

自从 Cooley-Tukey 提出上述 FFT 算法以来，人们又提出了不少新的快速算法，目的是进一步减少运算量以提高计算效率。其中重要的有 Rader-Brenner 算法、Z 变换算法、基-3 新算法、递归割圆分解算法、分裂基算法、余弦变换算法、Winograd 算法以及素因子算法等等。在多维 DFT 的快速算法方面，有以一维 FFT 算法为基础的行列算法和嵌套算法、向量基算法、多项式变换算法等。

参考文献

1. Nussbaumer H J. Fast Fourier transform and convolution algorithms. Berlin: Springer-Verlag, 1981

2. 蒋增荣，曾泳泓，余品能. 快速算法. 长沙：国防科技大学出版社，1993

（蒋增荣）

kuaisu yitaiwang

快速以太网（fast Ethernet） 采用带碰撞检测的载波侦听多址访问（CDMA/CD）方法进行介质访问控制的、传输速率为 100 Mb/s 的以太网。在以太网上运行的应用软件可毫无修改地在快速以太网上运行。快速以太网与以太网的主要区别是物理层上采用了新的信号编码方式，以便能在双绞线和光纤上以 100 Mb/s 的数据速率传送数据。

快速以太网支持多种传输介质类型：

100Base-TX 使用 2 对非屏蔽或屏蔽的 5 类双绞线电缆，最大网段长度为 100 m，物理拓扑结构为星型。

100Base-FX 使用二根光纤，按所使用的光纤类型和工作模式的不同，它的最大网段长度为 150 m、412 m、2000 m 或更长至 10 km，它支持全双工的数据传输。

100Base-T4 这是为了充分利用建筑物中已安装的 3 类非屏蔽双绞线电缆而定义的传输介质，它使用 4 对 3、4 或 5 类非屏蔽或屏蔽双绞线电缆，最大网段长度为 100 m。

100Base-TX 和 100Base-FX 采用 4B/5B 编码。100Base-T4 采用 8B/6T 编码。

快速以太网引入了全双工操作模式。全双工操作模式不仅可提高带宽（理论上数据速率可达 200 Mb/s），更重要的是可提高网络的最大地理范围。快速以太网的数据传输速率为 100 Mb/s，比以太网快了 10 倍。由于 CSMA/CD 介质访问控制方法有一个冲突窗口的限制（参见**以太网**），快速以太网的最大地理范围只有以太网的十分之一，限制了快速以太网的实际应用。1997 年快速以太网引入了全双工操作模式，该模式不再使用带冲突检测的载波侦听多址访问（CDMA/CD）方法进行介质访问控制，因而免除了冲突窗口的这个限制，拓展了快速以太网应用的地理范围。

参考文献

1. Quinn LB, Russell RG. Fast Ethernet. New York: John Wiley & Sons, 1998

2. Spurgeon CE. Ethernet: The definitive guide. O'Reilly & Associates, 2000

3. IEEE Computer Society. IEEE Std 802.3-2008 section 2. 2008

（方起兴 王能）

kuandai wangluo jieru jishu

宽带网络接入技术（broadband network access technologies） 将主干网连至最终用户，速

率高于兆位的接入网技术。当前主干网正向超高速和超大容量的方向发展,高达 400 Gb/s ~ 1 Tb/s 的波分复用系统已投入应用。另一方面最终用户大量使用的个人计算机处理能力也飞速增长。但连接主干网和最终用户的接入网仍处在窄带水平。因此,接入网的宽带技术成为发展宽带网的关键。下面介绍宽带接入的几种类型:

（1）不对称数字用户专用线(ADSL)　这是在无中继的用户环路上使用有负载电话线提供高速数字接入技术,对少量使用宽带业务的用户是一种经济快速的接入方法(参阅**数字用户专用线**)。

（2）高比特率数字用户专用线(HDSL)　这是在无中继的用户环路上使用无负载电话线提供高速数字接入技术,可实现在双绞线上高带宽双向传输。

（3）甚高速数字用户专用线(VDSL/或称 VHDSL)　这是在 ADSL 基础上发展起来的,可在很短的双绞铜线上传送比 ADSL 更高速的数据。其最大的下行速率为 51 ~ 55 Mb/s,传输线长度不超过 300m(参阅**数字用户专用线**)。

（4）混合光纤同轴电缆(HFC)　这是基于现有电缆电视(CATV)网基础上的光纤光缆和同轴电缆混合的宽带接入技术。HFC 网是宽带接入技术中最先成熟和进入市场的。HFC 在一个 500 户左右的光节点覆盖区可以提供 60 路模拟广播电视,每户至少两路电话,速率为 10 Mb/s 的数据业务(参阅**混合光纤同轴电缆**)。

（5）吉比特无源光网络(GPON)　随着网络 IP 化进程的加快和 ATM 技术的逐步萎缩,导致基于 ATM 技术的 APON 技术在商用化和实用化方面受阻,在这样的背景下全业务接入网联盟(FSAN)和国际电联(ITU)以 APON 标准为基本框架,重新设计了新的物理层传输速率和 TC 层,推出了新的 GPON 技术和标准。

（6）以太网无源光网络(EPON)　是一种将以太网技术与 PON 的传输结构结合起来,在 802.3 协议框架内制定的 IEEE/EFM 标准。EPON 采用点到多点结构的无源光纤传输方式,其下行速率可达到 10 Gb/s,上行以突发的以太网包方式发送数据流。另外,EPON 也提供一定的运行维护和管理(OAM)功能。(参阅**无源光纤用户线路**)

（7）宽带固定无线接入技术　主要有以下三类:①多路多点分配业务(MMDS),带宽为 200 MHz;②卫星直播系统(DBS),带宽为 500 MHz;③本地多点分配业务(LMDS),工作在毫米波段,大致在 28 GHz 附近,可用带宽至少为 1 GHz。宽带固定无线接入技术代表了宽带接入技术的一种新的不可忽视的发展趋势,它具有敷设开通快、维护简单、用户密度大时成本低的特点,是传统电信业务的有力竞争者。

参考文献

1. 韦乐平. 宽带接入技术的新发展. 电子科技导报, 1999(1)

2. 柯赓. 接入网技术与应用. 西安: 西安电子科技大学出版社, 2009

（程时端）

kuandai zonghe yewu shuziwang

宽带综合业务数字网(broadband integrated services digital network, BISDN)　指用户线上的传输速率在 2 Mb/s 以上的 ISDN。它是在 ISDN 基础上发展起来的,可以支持各种不同类型、不同速率的业务,不但包括连续型业务,还包括突发型宽带业务。其业务分布范围极为广泛,包括速率不大于 64 Kb/s 的窄带业务(如语音/传真)、宽带分配型业务(广播电视、高清晰度电视)、宽带交互型通信业务(可视电话、会议电话)、宽带突发型业务(高速数据)等。根据设计,宽带 BISDN 采用的传输模式主要有高速分组交换、高速电路交换、异步传输模式 ATM 和光交换方式四种。随着互联网的发展,有关 BISDN 的工作已经完全停止。

参考文献

程时瑞. 综合业务数字网. 北京: 人民邮电出版社, 1993

（马妍　马严）

kuangjia biaoshi

框架表示(frame representation)　将事物的各种属性、特征、变异及事物之间的类属关系等集成于框架中的一种结构化**知识表示**方法。

框架通常由描述事物各个方面的槽组成,每个槽描述框架所表示实体的一个属性。作为对槽的进一步说明,槽可以拥有多个侧面,从不同方面描述槽的各种特性。每个侧面又有一个或多个侧面值,它们可以是一个值或一个概念的描述。

一个框架的一般结构如下:

〈框架名〉

　　〈槽-1〉:　〈侧面-11〉〈侧面值-11〉

　　　　　　　……

　　　　　　　〈侧面-1m〉〈侧面值-1m〉

　　　　　　　……

〈槽-n〉:〈侧面-$n1$〉〈侧面值-$n1$〉

……

〈侧面-nm〉〈侧面值-nm〉

槽或侧面的取值可以是二值逻辑的真或假、实数值、文字以及其他形式的定义域,也可以是一组子程序,称为框架的程序附件。

观察事物时,外界事物、事件等在头脑中形成特定的概念,并在记忆中形成有关某物的整体结构,这一结构就称为框架。框架提供了一个对象或一类对象的结构化表示,可以描述典型状况、事实、对象和概念及它们之间的层次关系。这种表示既是层次化的,又是模块化的,是一种理想的结构化知识表示方法。用它来表示有关事物的知识时,不仅可以表示出事物各方面的属性,而且可以表示出事物之间的类属关系、事物的特征和变异等,在识别、分析、预测事物及其行为方面都有十分重要的意义。

1975 年,美国省理工学院 Marvin Minsky 首先提出了框架理论。该理论认为人们对现实世界中各种事物的认识均以一种类似于框架的结构存储在记忆中。当面临一个新事物时,就从记忆中找出一个合适的框架,并根据实际情况对其细节加以修改、补充,从而形成对当前事物的认识。因此,当遇到新情况或换个角度看问题时,通过比较老框架与新情况,把新的数据填入该框架形成一个特定概念,并且还可根据典型框架预测所属范围内的一个未来情况。在框架结构中填入有关新情况的数据时,可根据以往经验获得的概念对这些数据进行分析和解释。此外,框架结构也提供了在具体上下文中对特定实体进行预见驱动的处理方式,在此上下文环境下根据这种结构可寻找那些预见的信息。

在用框架表示知识的系统中,推理主要是通过框架匹配与填槽来实现的。框架系统中可使用多种类型的推理,如默认推理、存在性推理、公共属性推理、异常情况确认推理和类比推理等。一般框架推理过程如下:首先要用一个称为问题框架的框架结构来表示待求解问题;然后将初始问题框架与**知识库**中的框架进行匹配,如果两个框架对应的槽没有冲突或满足预设条件就认为这两个框架匹配成功;通过匹配找出一个或几个与输入信息最适合的预选框架,形成初步假设(即由输入信息激活相应的框架);在该假设引导下,收集进一步的信息;最后按某种评价原则对预选框架进行评价,以决定是否接受该预选框架(即在框架引导下的推理)。

综上所述,框架表示法为概念、结构和功能模型等陈述性知识的描述提供了一种结构化的典型方法,但其对过程性知识的表达能力比较差,将框架表示与**产生式表示**结合在一起可望较好地解决这一问题。

参考文献

1. Minsky M. A framework for representing knowledge: The psychology of computer vision. New York: McGraw-Hill, 1985

2. 陆汝钤. 人工智能(上). 北京:科学出版社,1989

(申富饶　陈世福)

L

languang guangdie qudongqi

蓝光光碟驱动器(blu-ray disc drive,BDD) 一种能读取蓝光碟的读写装置。采用的主要技术与 CD、DVD(参见**数字多用途光碟**)类似,只是波长更短(从 CD 的 780 nm、DVD 的 650 nm 减短到蓝光波段的 405 nm)、数值孔径(又称镜口率)更大(从 DVD 的 0.60 增大到 0.85)。由于利用波长较短的蓝色激光读取和写入数据,并因此而得名。向下兼容 DVD、VCD、CD 等格式,简称蓝光光驱。蓝光碟的英文名不使用"Blue-ray disc"的原因,是"Blue-ray"这个词过于通俗、口语化,不能构成注册商标申请的许可,因此蓝光光碟联盟去掉 blue 中的英文字母 e 来完成商标注册。

蓝光光碟是继 CD、DVD 以后的第三代光碟产品。如果说 DVD 是为播放标准清晰度的视音频节目设计,那么蓝光光碟就是为播放高清晰度的节目而设计。蓝光光碟同时采用了更高效率的信道与信源的编解码技术和热化学反应模式代替传统的光化学反应模式,使母碟刻录可以获得比聚焦光斑更小的记录斑,以及在读信道处理技术上采用了近几年来最新发展的 PRML(部分响应最大似然)等技术,蓝光光碟的数字版权管理系统(DRM)也更加完善可靠与可追溯。2003 年,世界上推出的首款蓝光光碟机是由日本索尼公司制造的蓝光光碟录像机,紧接着蓝光光碟播放机、蓝光光碟驱动器等产品相继面世。

目前,世界上主流的蓝光光碟是 BD(blu-ray disc),此外,还有 HD-DVD 等;蓝光光碟的容量从 25 ~ 200 GB;读取速度从单倍速(1X)36 Mbit/s 到 6 倍速(6X)216 Mbit/s;主要用于数据存储、高清晰度的影视节目、高档游戏机等。

参考文献

1. 徐端颐. 高密度光盘数据存储. 北京:清华大学出版社,2003

2. 中国电子技术标准化研究所. 2009—2010 信息技术标准化指南. 北京:中国标准出版社,2011

(潘龙法)

lanya PAN

蓝牙 PAN(Bluetooth PAN) 蓝牙是一种用于固定和移动设备之间的短距离数据交换的无线技术,能够构建高安全性的个人区域网络,蓝牙 PAN 也被称为 piconet,它最多由 8 个 WPAN 设备以主从关系组成。Piconet 中的第一个蓝牙设备起到主导的作用,其他的设备都从属于它并与它相通信。尽管在理想情况下 Piconet 的覆盖范围能够达到 100 m,但是在现实中一般为 10 m。经过多年的研究与开发,蓝牙技术也发展出了多个版本,从最初的蓝牙 1.0 到现在的蓝牙 3.0 技术期间,蓝牙技术得到了飞跃式的发展。

在蓝牙技术的基础上,IEEE 协会提出了基于蓝牙无线技术的标准——IEEE 802.15。IEEE 802.15 工作组于 1998 年成立,专门从事 WPAN 标准化工作,它的任务是开发一套适用于无线个人区域网络的标准。IEEE 802.15 标准总共包括 5 个任务组,其中 IEEE 802.15.4 协议得到了人们的广泛认可。Zigbee 技术就是由 IEEE 802.15.4 协议发展而来的。

(周正)

lei

类(class) 一组具有共同特性的相似对象的抽象描述。类是**面向对象语言**的基本成分。它可进一步理解为:①面向对象程序的唯一构造单位;②抽象数据类型的具体实现;③对象的生成模块。类的出现,为面向对象编程的三个最重要的特性(封装性、继承性、多态性)提供了实现手段。

20 世纪 60 年代第一种面向对象语言 Simula67 最早引入了类的要领和继承,成形和完善于 70 年代的 Smalltalk。

面向对象程序由一组相关的类构成,故类是静态的;程序的执行体现为一组相互通信的对象的活动,故对象是动态的。类是一组具有共同特性的相似对象的抽象描述,对象是其具体实例。

类的成员分为数据成员和成员函数。数据成员指定了该类对象的内部表示,成员函数指定了该类

的操作。一般而言,类成员分为公有、私有和保护型。公有成员可在类外访问;私有成员只能被该类的成员函数访问;保护成员只能被该类的成员函数或派生类的成员函数访问。数据成员通常是私有的,成员函数有公有和私有之分。公有函数可以在类外被访问,也称之为接口,可为各个数据成员和成员函数指定合适的访问权限。

①按定义方式分为系统类和用户类:前者由系统内部定义;后者由用户自行定义;②按继承关系分为基类和衍类;后者通过继承前者的属性和操作而定义。衍类也称子类,基类也称父类。需要由其衍类进一步定义的类称为延迟类或抽象类,抽象类被定义为永远不会也不能被实例化为具体的对象,用于定义一种抽象上的概念,在类的继承关系中它往往被定义在较上层的位置。

类间的主要关系是构成类间层次结构的继承关系。

在面向对象语言中,类的作用有二:①作为对象的抽象描述机制,类刻画相似对象的共同属性和行为;②作为程序的基本构造单位,类支持模块化设计,其分类关系是模块划分的规范标准。

在面向对象语言中,类的益处有二:①信息隐藏,指类封装其内部包含的特定信息,对于不需要这些信息的其他类来说,是透明的;②可复用性,指程序员可用相同的类重复构建对象。

传统的面向对象语言 Smalltalk 和 Eiffel 及现代主流编程语言如 C ++ 、C#和 Java 等都支持类的概念。这些编程语言在支持与类相关的各种特性方面都有一些微妙的差异,大多数都支持不同形式的类继承,许多语言还支持封装性和多态性的特性。

参考文献

1. 徐家福,王志坚,翟成祥. 对象式程序设计语言. 南京:南京大学出版社,1992

2. Stephen S. Object-oriented and classical software engineering. 7th ed. McGraw-Hill. 2006

(王志坚　张鹏程)

leibi xuexi

类比学习(learning by analogy)　通过问题的相似性,基于以往问题(源问题)的解决方案,来求解新问题(目标问题)的学习方法。

类比学习早期的理论基础来自认知科学(cognitive science)。其中的结构映射理论强调,源问题和目标问题之间的共有结构,可以通过某种映射(mapping)来学到。而这种映射,既可以建立在物体之间,也可以建立在领域之间。同时,学者还提出结构相似性的最大化,认为在两个领域之间存在正、负等多元制约。当制约满足最大化的时候,这两个领域将达到结构的一致性,继而产生学习和推理的结果。

类比学习的方法主要包括问题之间的相似度学习,**基于案例的推理**系统(case-based reasoning, CBR)和迁移学习(transfer learning)。

在相似度学习方面,学者提出利用动态记忆(dynamic memory)对以往的案例进行组织和改进,从而学习和解决新的问题。其中一种方法,是把学习问题表达成图形之间相似度度量的问题。之后,很多的学习系统将这一概念标准化,使得类比学习成为一些著名**人工智能**系统的核心。比如,在解释学习(explanation based learning)领域著名的 PRODIGY 系统,就利用类比学习的原理来对新的规划问题进行修改,从而更快地得到新问题的解答。又如,SOAR 系统利用已经学到的知识,通过知识模块(chunking)之间的对比来达到学习效果。

基于案例的推理(CBR)系统是类比学习的一个具体实现。这一学习方法是把学习问题划分为四个主要阶段:抽取,再利用,修改和保留。早期的 CBR 系统 CYRUS,可以用来回答有关行程的问题。CYRUS 的主要贡献在于将实例的记忆和抽取规范化。中间人(MEDIATOR)系统可以帮助两个利益集团解决现有的冲突。烹饪(CHEF)系统在规划算法的基础上,通过分析旧实例中的错误,发展出新的菜谱。

类比学习也强调在不同问题之间进行知识迁移,使得过去学习得到的知识能够在新领域中得到再利用。迁移学习注意到不同数据在分布、表达和结构上的区别,加以**统计学习**,建立可靠的映射关系,从而实现对目标数据的学习。迁移学习在同结构与异构空间下,利用统计学习进行基于案例或特征的知识迁移,是类比学习的扩充。

参考文献

1. Gentner D. Structure-mapping: A theoretical framework for analogy. Cognitive Science, 1983, 7: 155-170

2. Falkenhainer, et al. The structure-mapping engine: Algorithm and examples. Artificial Intelligence, 1989, 41: 1-63

3. Pan S J, Yang Q. A survey on transfer learn-

ing. IEEE Transactions on Knowledge and Data Engineering, 2010, 22(10): 1345-1359　　(杨强)

leiren jiqiren

类人机器人(humanoid robot)　模仿人的形态和行为的**智能机器人**,亦被称为仿人机器人。它不仅外观像人,还能像人一样活动,甚至会像人一样地思考问题,有一定的智商和情商。事实上,很难严格地定义类人机器人,从外形看一般认为,类人机器人有躯干、头、双臂和双腿,有的还有脸、眼和嘴,有的只有腰以上的部分。

类人机器人系统分为传感、驱动、规划和控制四个部分。从物理、认知和社会性等各个层面模仿人的外观和功能,但需要特别指出的是类人机器人终究不能成为人。类人机器人具备的基本功能包括:自我维持(如自动充电)、自主学习(指在没有外界帮助下学习或获取新的知识,适应环境变化的能力)、趋利避害,以及与人和环境安全交互。

类人机器人涉及心理学、生理学、生物学、神经和认知科学甚至哲学、语言学等领域,集机械、电子、材料、计算机、人工智能、机器人和控制技术多门学科于一体,有着十分宽泛的研究题材,吸引了广泛的关注。有的研究机械硬件设计,有的注重它模仿人的认知过程,有的更在意它在服务和娱乐行业的商业用途。概括来讲,主要集中在下面的几个研究方向:

感知:包括计算机视觉,味觉、嗅觉、声呐、红外、触觉、距离等各类传感器信息处理技术,以及诸如前庭视觉反射的各种潜意识生理机制,涉及传感信息融合和感知信息分类。

机器人与人交互:研究控制和执行任务过程中的人为因素,实现机器人与人的有效交互。由于类人机器人体积大又重,交互过程人身安全要有保证。目前机器人与人的交互仅限于固定的单词以及简单的手势和脸部表情交流,交流的深度和广度都有待于提高。

自适应学习:类人机器人的智能来源于学习,使它能够调整自己的行为应对环境发生的变化,并组合已有行为能力学习新的任务。机器人的学习能力受制于获取的传感信息、学习方式和训练时间。作为常规的学习算法,监督和无监督的机器学习能够产生鲁棒的、普遍性的行为。学习是实现类人机器人智能控制的一项重要任务。这样,机器人接收的指令将是"做什么",而不是告诉它"怎么做"。类人机器人的行为计算模型,将模拟人类通过感官和运动神经实现传感信息学习的机制。

直立行走:在粗糙不平的地形和很陡的斜坡上行走,或者上下楼梯,这些对类人机器人而言是一个艰难的任务。它的问题不仅在于腿的向前或向后机械运动,而在于整个机器人身体的快速动态平衡能力。

手臂灵巧作业:现今的类人机器人已经能够完成一些灵巧度很高的操作,比如抓球、玩杂耍、跳舞、远程外科手术和倒咖啡等。类人机器人手臂研究有明显的进步,现在的手臂更轻便、圆滑、柔顺,手指也更加灵巧。然而,完成灵巧操作的重点还在于手臂的自适应控制。

类人机器人将适应人类的生活和工作环境,扩展人类的能力,广泛应用在服务、康复医疗、娱乐等领域。它体现一个国家机器人技术的综合发展水平,因而,许多发达国家不惜投入巨资进行相关的开发研究。最早的类人机器人研究来自日本,在1973年,早稻田大学加藤一郎教授就开始了这个领域的研究。在2000年,本田公司推出了ASIMO(如图1)。它是世界最先进的类人机器人之一,所具备的功能不断地冲击人们的想象,能够完成快速奔跑和非常复杂的体操和跳舞动作。COG是美国MIT研制的类人机器人,用来模拟人的各种感觉器官,包括视觉、听觉、触觉、本体感受和前庭系统。德国DLR研制的Justin是一款轮式移动的类人机器人,通过视觉和触觉等,它能够完成不亚于人的精细和敏捷的动作,比如接抛球,冲咖啡等。我国在类人机器人研究领域里也做了大量工作,且在很多方面取得了重大的进展。



图1　本田公司推出的类人机器人

参考文献

陈恳,付成龙. 仿人机器人理论与技术. 北京:清华大学出版社,2010

(杨唐文)

leixing dingyi

类型定义(type definition) 程序设计语言中的类型扩充设施。程序人员使用这种设施可以自行定义所需的类型。在有些语言中类型定义又称**类型说明**。类型定义的作用有二,一是起缩写作用,二是定义新类型。

类型定义的一般形式是:

type T = 类型

这里的类型为已知类型或其中又包括所定义的类型T,前者起缩写作用,后者定义新类型。

例 1:

```
type person = record
                  name: packed array [1..20] of
                  char;
                  age: integer;
                  height: real
               end
```

例 2:

```
type TP = record
             f1: char;
             f2: TP
          end
```

(徐家福)

leixing lilun

类型理论(type theory) 为避免集合论悖论而建立的,主要研究集合的分层、分类方法(包括公理化方法)的数学理论。

近20年来,它在计算机科学中得到广泛应用。20世纪初B. Russell提出了类型论作为公理集合论乃至数学的基础,30年后A. Church用λ演算和类型的概念定义了高阶逻辑的形式系统。但在经典数学中,类型理论始终未受到广泛关注,直到60年代末才有大的改观。原因有三:第一,20世纪初可构造性数学虽成为数学的一独立分支,但由于W. Brouwer开创的直觉主义并没有得到很多支持,一般人的信念是数学的主要部分不可能构造地研究,改变这一信念的是E. Bishop的工作,构造性地重建了经典分析的核心部分,表明了构造数学是可以与经典数学相媲美的。60年代E. Bishop及其追随者在构造数学领域中作出的重要成果使直觉主义得到发扬光大。直觉主义逻辑的证明论启用了将类型作为基本概念的方法。基本思想是将一个可构造命题看成类型,即类型的元素为该命题的一个证明。这便是"命题当作类型"观点(柯里-霍华德观点,或称解释)。J. Y. Girard首先使用该原则给出二阶直觉主义逻辑的类型表示,并证明该逻辑是典范化的。P. Martin-Löf也提出了他的类型理论,为可构造性数学提供了基础。第二,TOPOS理论的出现,建立了TOPOS的内逻辑与直觉主义逻辑两者间的联系,在更广泛的范围内揭示了范畴结构的逻辑特征,对类型理论的认识更深入了。第三,也是最重要的是计算机科学的推动作用,高级程序语言不断涌现,相应的类型系统也得到研究。另一方面,在计算机上实现了用类型论求解数学问题的系统。

以"类型理论"名称出现者,颇为广泛,有分枝类型论、简单类型论(罗素)、T理论(K. Gödel)、F理论和Fω理论(J. Y. Girard)等,其中与计算机科学有关的有以下几种:

(1)逻辑类型理论,它包含内逻辑。该理论目的是为构造数学提供类型论基础。通常所说的构造演算即是逻辑类型系统。本质上是高阶直觉逻辑的类型表示。逻辑类型系统的著名的子系统有:多态λ演算,马丁洛夫类型理论和逻辑框架。

(2)程序类型理论,研究程序语言的类型系统。新型程序语言、程序设计方法学、语义理论的迅速发展迫切需要类型系统的研究。使用"类型"这一概念大致有两个含义:一指论域的多种组合形式,二指语法范畴的分层结构。该理论涉及的类型研究包括下述基本内容:(表达式a有类型A,记为a: A,读作a为A的一个居元。反过来也说A有居元a。)

不动点操作:设A为类型,不动点算子Y: $(A\to A)\to A$,使得对于每个f: $A\to A$,有Yf: A且$Y(f)=f(Y(f))$。不动点算子是引入不终止计算的基本工具。

归纳类型:类型为A的变量x是正定的是指x只出现在偶数个(函数类型)符号→的左面,例如x在$x\to y\to z$中即为正定的。归纳类型产生规则:A为类型且x在A中是正定的则$\mu x.A$为类型。许多数据结构如自然数、表、树等均是归纳定义的。

递归类型:归纳类型产生规则中删去正定条件即为递归类型。它与归纳类型差别在于前者是非良序的,后者是良序的。分别加入到二阶λ演算中,加入后者时将是强典范化的,递归类型则不然。

多态类型：一程序处理不同类型的输入，有两种多态类型：特定型和参数型。特定型对不同类型的输入反应是任意的无规则的；参数型对不同类型输入的反应是统一的。例如，项 t 有参数型 $\forall x$: type. $A[X]$，t 看成函数，对于类型为 T 的输入给出结果为项 $t(T)$，其类型为 $A[T/X]$。

记录类型：记录是程序设计语言的中心概念。对象是记录，属性是记录命名值。记录类型是一组命名类型。例如，记录〈x = True，y = 2〉有类型《x: Bool，y: Int》。

子类型：直觉上说，类型 A 是类型 B 的子类型当且仅当 A 的每个居元是 B 的居元，记为 $A<B$。这个子类型关系满足自反、传递性质。关于函数类型操作有

$$\frac{A < A', B < B'}{A \to B < A' \to B'}$$

意即类型 $A \to B$ 的程序必为类型 $A' \to B'$。继承性在类型论上较为贴切的描述是使用子类型关系。

交类型：对任意两个类型 A，B 可定义交类型 $A \wedge B$。无论从证明论、模型论角度看，交类型都是较艰涩的概念。例如设项 f 为类型 $A \wedge (A \to A)$，则 $f(f)$ 便有定义。因为交类型 $A \wedge (A \to A)$ 既是 A 的子类型又是 $A \to A$ 的子类型，故有 f: A 及 f: $A \to A$，这种自身作用在自身上的项，在一般类型理论中不能合适地定义其类型。

高阶类型：以其他类型作为其居元的类型。例如，U_0: U_1 可解释为：如果 U_0 代表程序的类型，则 U_1 就是程序的规约的类型。

至今已有不少类型系统问世，它们的证明和模型的研究都很有意义。现行的程序设计者总是在追求新的程序设计思想，因此类型的研究是一个具有活力的方向。

参考文献

1. Howard W. The formulae-as-types notion of consturction. In: Seldin J P, Hindley J R, eds. To Curry H B: Essarys on Combinatory Logic, Lambda Calculus and Formalism. New York: Academic Press, 1980

2. Martin-Löf P. A theory of types. Report 71～73, Dept. of Mathematics, University of Stockholm, Feb. 1971, Revised Oct. 1971

（陆汝占）

lisan shijian xitong fangzhen

离散事件系统仿真（discrete event system simulation）　对离散事件系统建立数学模型，并在计算机上对模型进行试验，进而进行系统分析的仿真技术（参见**计算机仿真**）。**离散事件系统**是指由于随机事件的驱动使得系统的状态只在一些离散的时间点上发生变化的系统。

离散事件系统的状态量由随机事件的驱动而发生变化。在两个相邻事件发生之间，系统状态量是保持不变的，即是离散变化的。由于离散事件系统固有的随机性，对这类系统的研究分析往往比较困难。经典的概率及数理统计理论、随机过程理论虽然能对一些简单系统提供解析解，但对大量的实际系统，仍需运用仿真技术来提供较为满意的结果。离散事件系统仿真技术是针对各种离散事件系统所共有的一些特性而产生、发展并得到广泛应用的。

图 1 表示离散事件系统仿真的一般步骤。下面就各步骤的一些特殊问题加以说明。

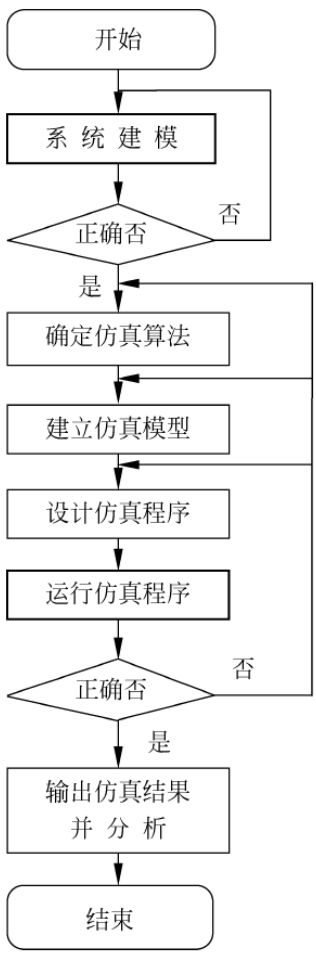


图 1　离散事件系统仿真的一般步骤

（1）系统建模　一般用流程图的形式加以描述，重点描述临时实体的产生规律，历经系统的过程，永久实体对临时实体的作用规则、条件及结果。

随机变量模型的确定在离散事件系统建模中占有十分重要的地位,要搜集足够多的原始数据以便较好地确定其分布的类型及参数。对于系统模型应进行有效性检验,以确定所建系统模型能有效地代表所研究的真实系统。

(2) 确定仿真算法　主要包括两方面的内容,即从所要求随机变量模型中产生相应的随机变量以及选择建模策略(参见**离散事件系统仿真建模方法学**)。

(3) 建立仿真模型　根据已确定的仿真算法,建立被仿真系统的计算机模型。仿真模型是系统状态转移的动态描述,因此首先要定义系统的状态变量,这要根据系统的内部结构及仿真研究的目的来确定。即使是同一系统,仿真研究的目的不同,系统状态定义也可能不尽相同。在离散事件系统中,状态的变化是由事件引起的,因此,要在定义系统状态的基础上定义系统事件及其有关属性。系统事件的定义与建模策略有关,如采用事件调度法建模,则要定义出系统中的全部事件及相应的事件处理流程,包括事件类型、事件发生时间、同时发生事件的处理规则等。如采用活动扫描法及进程交互法,则还要定义相应的活动或进程,以便按活动或进程的观点来建立仿真模型。

(4) 设计和运行仿真程序　仿真模型最终是通过仿真程序来实现的。可根据所拥有的仿真工具的情况选择某种高级语言(如 FORTRAN,C 等)或离散事件系统**仿真语言**来实现所建立的仿真模型。在仿真试验前,首先要校验仿真程序的正确性,然后校验仿真模型的正确性——经过验证是正确的仿真模型才能进行仿真试验。仿真试验条件及运行次数与输出分析的要求有密切关系,这是与连续系统仿真的重要的区别。

(5) 分析仿真结果　对具有随机性因素的离散事件系统模型的一次运行仅仅是随机过程的一次抽样,仿真结果的置信度应予以特别的注意。

参考文献

肖田元,范文慧. 系统仿真导论. 2 版. 北京:清华大学出版社,2010

(肖田元)

lisan shijian xitong fangzhen jianmo fangfaxue

离散事件系统仿真建模方法学(modeling methodology of discrete event system simulation)　以某种观点(如事件、活动或/和进程)描述离散事件系统并建立相应仿真模型的方法的总称。**离散事件系统仿真**的核心问题是建立描述系统行为的仿真模型。虽然离散事件系统大多是随机系统,但由于仿真模型中采用的是伪随机数,从理论上讲,其状态的转移也是确定的,因而也可得到确定性的状态转移函数。在一个较为复杂的离散事件系统中,一般都存在诸多的实体,这些实体之间相互联系,相互影响,然而其活动的发生统一在同一时间基上,采用何种方法推进仿真钟,建立起各类实体之间的逻辑联系,这是离散事件系统仿真建模方法学的重要内容,有时称之为仿真算法或仿真策略。

目前,比较成熟的有四种仿真建模方法,即事件调度、活动扫描、进程交互和三阶段法。在建立仿真模型时,一般并非只拘泥于采用某一种策略,同一个仿真模型有时可同时采用几种策略。下面对上述四种方法加以说明。

事件调度法　事件调度法(event scheduling)用“事件”的观点来抽象真实系统(参见**离散事件系统仿真**),即通过定义事件及每一事件发生对系统状态的影响,并按事件发生时间顺序来确定每类事件发生时系统中各实体之间的逻辑关系及其状态,这就是事件调度法的基本思想。

采用事件调度法建立仿真模型时,所有事件均按时间先后存放在事件表中;同时,模型中要设计一个时间控制部件,该部件的作用是实现仿真钟的管理与控制,即每当处理一类事件时,它总是从事件表中选择具有最早发生时间的事件,并将仿真钟推进到该事件发生的时间,然后调用与该事件相应的事件处理模块。任何一个事件处理模块在执行完后都必须返回到时间控制部件。这样,事件的选择与处理不断地进行,仿真钟按事件时间往前推进,直到仿真终止的条件满足为止。

这种方法的特点是仿真钟的推进仅依据事件发生的时间,因而,在建模时有两个基本问题需要加以特别注意。第一是所谓“同时事件”,即具有相同发生时间的事件,模型中必须事先规定其处理顺序,亦称为规定“解结规则”。这一般是通过定义事件的优先级来解决的。第二是所谓“条件事件”。从本质上讲,事件调度法是一种“预定事件发生时间”的策略,如果按预定时间某一事件应该发生,但发生该事件的条件(如果有的话)不满足,则必须推迟或取消该事件的发生。所有上述两方面的问题,都应在相应的模块中特别加以处理,以免产生模型的死锁。

活动扫描法　活动扫描法(activity scanning)用

“活动”的观点来描述实际系统，即通过定义活动及每一活动发生对系统状态的影响来建立系统模型。所谓“活动”就是有关联的两个事件之间的过程。这种建模策略特别适用于对活动持续时间有较强不确定性的系统进行仿真。由于是采用“活动”的观点建模，活动扫描法要求定义系统中所有“活动”及相应处理“活动”的模块，包括定义活动发生的条件，而活动发生的时间也作为条件之一，只不过它是具有最高优先权的条件。同时，这种建模策略包括了两个控制仿真流程的基本部件，即活动扫描部件及条件处理部件。活动扫描部件的任务是在仿真的每一步对系统中定义的全部活动按优先级从高到低逐个扫描；而条件处理部件则用于对被扫描活动的有关条件进行测试。

三阶段法 三阶段法（three phase method）结合了事件调度法和活动扫描法的特点，也称为扩展的活动扫描法。它将整个仿真控制过程分为三个阶段（参见图1），程序实现也颇为简练，因而在仿真应用中被大量采纳。

在三阶段法中的A阶段进行时间扫描，即扫描事件表，找出下一最早发生事件，将系统仿真钟推进到该事件的发生时刻，系统时钟一直保持这一时刻直到下一个A阶段发生。具体做法是：仿真控制程序搜寻出那些“可用性”属性为“假”且具有最小时间片（time cell）的实体，并将该time cell作为下一最早事件的发生时刻。注意，因为有可能有多个B活动在下一时刻发生，所以仿真控制程序必须记录在该时刻所有的不可用实体而形成一个DueNow列表。

在B阶段执行DueNow列表。一旦DueNow列表形成，仿真控制程序将顺序扫描列表中的每个实体，从中挑选出可执行的实体，对于每一个可执行实体进行如下操作：将实体从DueNow列表中删除；将该实体的“可用性”属性置成“真”；执行该实体“下一活动”属性所代表的活动。

在C阶段，查询C_s事件表。逐一对其中的事件进行条件测试，看看其条件是否满足。如果条件满足，则执行相应的动作。在查询C_s事件表期间，保持当前仿真钟不变，直到所有的C_s事件都不满足启动条件。

进程交互法 进程交互法（process interactive）用“进程”的观点来抽象真实系统。它将实体历经系统时所发生的事件及活动按时间及逻辑顺序进行组合，从而形成各种进程。这种方法主要用于对实体

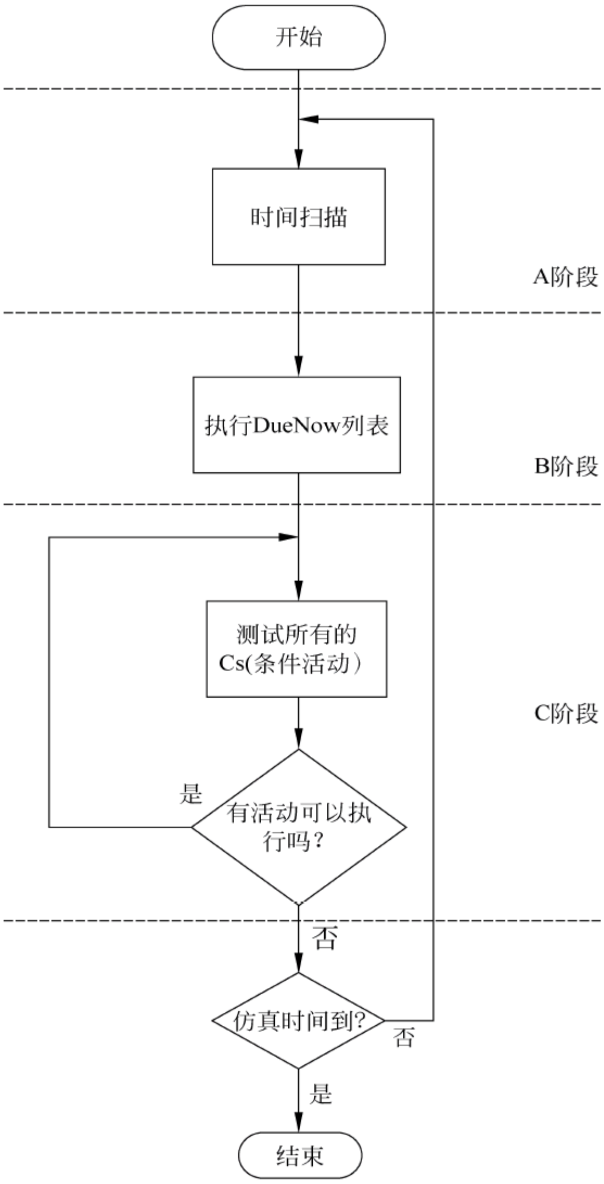


图1 三阶段法框图

活动较规则的系统建模。

这种策略要求建立两种事件表，第一种称为当前事件表（CEL），它存放着从当前时间点开始有资格执行的某一进程中的某一事件记录，该事件是否能发生尚未判断；第二种称为将来事件表（FEL），它包含在将来某个仿真时刻发生的某一进程中的某一事件记录。每一个事件记录中有若干个属性项，其中必须包括说明该事件所在进程及在该进程中的位置指针。

参考文献

1. James W H. Strategy related characteristics of discrete event languages and models. Simulation, 1986, 46(4): 152-159

2. Robert M D. The three phase approach: a comment on strategy related characteristics of discrete event and models. Simulation, 1986, 47(5): 208-211

3. 肖田元,范文慧. 系统仿真导论. 2 版. 北京:清华大学出版社,2010 (肖田元)

lisan shijian xitong fangzhen shuchu fenxi

离散事件系统仿真输出分析(output analysis of discrete event system simulation) 涉及离散事件系统仿真的实验设计以及运行结果的置信度分析。单一系统模型输出分析方法有两种,即:终止型仿真(terminating simulation)和稳态型仿真(steady-state simulation)。

终止型仿真是指在规定的时间内进行仿真并统计系统性能,仿真运行长度是事先规定的,因而系统的初始状态设置对仿真结果的影响不能忽略。为正确获得系统的统计性能,仿真实验设计要求多次独立运行仿真模型。为提高独立性,每次运行应该采用不同的伪随机数据流产生随机变量,至少在每次运行时随机数发生器采用不同的种子值,进而用经典统计分析方法来构造系统性能的置信区间。

稳态型仿真的目的是要估计系统的稳态性能,对仿真运行的长度没有限制。在稳态仿真中,为减少由于人工和不符合实际的初始条件引起的响应变量点估计偏差有两种解决方法:如果有实际系统,则收集该系统的数据,并利用这些数据指定更加典型的初始条件;如果不存在实际系统,可以将仿真分为两个阶段,首先是预热阶段,不进行响应变量的收集。当系统状态接近系统稳态行为时才进入第二阶段,即数据收集阶段。如果说系统已经到达一个近似的稳态,那么可以忽略响应变量点估计的偏差。

终止型与稳态型根据其对置信区间的精度要求不同,有两类基本的输出分析方法,即固定样本长度法及序贯法。

终止型固定样本长度法由用户规定独立运行的次数,每次运行的结果满足独立同分布的条件,而且是正态随机量。因此,仿真运行次数不能太少,否则运行结果的分布可能不对称。终止型序贯法是为得到有规定精度的置信区间的方法。先按固定样本长度法运行若干次,若得到的绝对精度或相对精度值太大,可再增加运行次数,直到满足要求为止。

稳态型固定样本长度法也称批均值法,是将仿真输出数据分批,得到每批数据;分别求得每批的均值,进而得到总的样本均值及其置信区间。稳态型序贯法则基于批均值法中任意两批数据之间相关系数的大小判断解决批长度、仿真运行总长度的确定,以得到能满足规定精度的置信区间。

参考文献

1. Averill M Law. Simulation modeling and analysis. 4th ed. New York: McGraw-Hill, 2000

2. 肖田元,范文慧. 系统仿真导论. 北京:清华大学出版社,2010 (肖田元)

lisan shuxue

离散数学(discrete mathematics) 研究离散对象数学结构及其性质的有关数学分支的总称。相对于实数理论、数学分析、微分方程这些研究连续对象的数学分支而言,离散数学以处理离散对象为特征。研究内容通常包括(但不限于)图论、**集合论**、组合数学、数理逻辑,以及各种代数结构等可以离散化或者枚举计数的数学对象。

离散数学的发展在很大程度上受到计算机科学的影响。因此离散数学主要的应用领域也是计算机科学,其中的概念和理论已经成为计算机科学的基础内容。如图论应用于网络、操作系统和编译程序,集合论应用于软件工程和数据库,组合学应用于算法和复杂性分析,逻辑学运用于软件工程和人工智能,代数结构则用于计算机体系设计、程序语言理论和网络规范描述。反过来,计算机科学的发展也推进了离散数学的深入发展,并且延拓着其研究领域。

图论本身是数学的一个分支,同时也被认为属于离散数学的研究范围。一个图是由顶点和边组成的离散结构。由于计算机科学和工程中很多问题可以抽象成图的形式,因此图的研究在计算机领域中十分重要和普遍。特别是在网络的研究中,根据用户和网站的关系,可以抽象成图的结构,进而通过对于图的研究来获取许多关于网络行为和用户行为的规律。图本身的一些数学性质的研究,例如最大团问题,最短路径问题,最小支撑树问题也为网络设计和分析提供了坚实的数学基础。

集合论是研究集合的数学理论,一般地,把集合论归于离散数学的范围。作为现代数学基础的集合论也是计算机科学的基本描述语言。集合论研究集合以及不同集合之间的相互关系。通过对于描述集合的定义语言,可以推断集合元素所具有的属性,以及集合的整体性质,这就是语义和语法之间的深刻关系。通过科学的抽象,使用集合论的语言来表述问题和概念,是形式化方法的基本特征,对于计算机

科学具有本质性的意义。集合论的内容和方法已经渗入到计算机科学的所有方面,特别地,在程序设计和数据库的研究和工程中,大量应用集合论的语言和结果。

组合数学是一个十分古老的数学分支,研究对象的组合性质及其状态的计数问题。从计算机的角度,这些组合状态与算法设计和分析、程序设计等领域有着密切的联系。典型的组合问题有四色问题、幻方问题、中国邮路问题和 Ramsey 问题等。

数理逻辑是研究逻辑公式及其相互之间的关系,主要是研究公式之间的推理关系。除了传统的逻辑学范围外,计算机科学也使用一些非传统逻辑的推理形式,例如模态逻辑、非单调逻辑等哲理逻辑学。无论是传统的逻辑推理还是非传统的逻辑推理,都在计算机科学,特别是人工智能领域有着基础性地位,是支持计算机学科发展的重要数学基础之一。数理逻辑主要研究逻辑命题的内容(语义)和获取方式(语法),从传统的逻辑来看,这种获取是在一组假设的公理下,通过约定的推理规则,由假设条件(逻辑前提)得到结论(逻辑后承)。但是在计算机科学领域中,这种获取的方式还包括机器学习、语义变换等非传统的形式,这些方法在模式识别、数据分析、系统建模、情感计算等领域有着广泛的应用(参见**逻辑学**)。

代数结构是研究代数系统的结构问题,在离散数学中,通常是指可数系统。由于几乎所有的系统都可以通过定义一些运算而成为代数系统,因此代数系统具有的广泛适用性。离散数学主要研究各种代数系统的结构问题,这些研究为计算机科学中的算法设计、系统分析和问题求解提供了数学基础。很多计算机科学中的问题,都是将研究对象抽象成为一个代数系统而得到深入的研究。例如形式语言与自动机就是一个典型的代数系统,其数学的表达模式已经成为熟知的内容,其中最著名的图灵机则成为理论计算机中研究可计算性与计算复杂性的重要工具。另外在通信并发系统,进程代数已经成为一种有力的研究工具,通过公理化的描述,揭示了通信过程中变迁、模拟、测试等方面的深刻性质和相互关系。

参考文献

1. 屈婉玲,耿素云,张立昂. 离散数学. 3 版. 北京:高等教育出版社,2009

2. Rosen K. Discrete mathematics and its applications. 6th ed. New York: McGraw-Hill Companies, 2011

(金小刚)

lichujue fankui zhuangzhi

力触觉反馈装置(haptic device)　一种能够产生力触觉反馈信号的人机交互装置。在**虚拟现实**系统中,操作者通过力触觉反馈装置与虚拟场景交互,感知虚拟物体的粗细、软硬、冷暖、轻重等物理特征,通过与视觉、听觉等感觉通道的配合,能够有效增强操作者的**沉浸感**。

人类获取力触觉信息的通道可以分为皮肤和肌肉两类。相应地,力触觉反馈装置也可分为两大类:触觉再现(tactile display)和力反馈(force feedback)。通过触觉再现装置,操作者能够感知虚拟物体的纹理、粗糙度、形状、温度等属性;通过力反馈装置,操作者能够感知虚拟物体的重量、惯性、刚度,以及接触力和摩擦力等。

力反馈技术在 20 世纪 50 年代应用于核环境下的遥控操作,使操作者在处理核原料时免受核辐射。70 年代后期,虚拟现实技术的出现推动了人机交互技术的发展。90 年代初在虚拟现实仿真系统中提出了触觉反馈的概念,力反馈技术第一次让用户感受到了虚拟物体的纹理。90 年代中期第一台面向虚拟环境应用的商用力反馈装置 PHANTOMTM 面市。经过多年的发展,力反馈技术已日趋成熟。

力反馈装置的控制方式分为阻抗和导纳两种(图 1)。阻抗型装置是通过输入位移或速度计算输出力,而导纳型装置是根据输入力计算输出位移或速度。由于采用阻抗控制的装置设计简单,制造便宜,目前大多数力反馈装置都是阻抗型。

力反馈装置根据安装位置的不同可分为两大类:一类安装于桌面或机架,美国 SensAble Technologies 公司的 Phantom 系列是典型的桌面式装置。另一类可佩戴在操作者身上,主要有外骨架装置和**传感手套**。外骨架装置能够为操作者的上肢或下肢提供力反馈信息,数据手套能够使操作者感受到抓取虚拟物体时的作用力。

触觉再现装置是通过刺激皮肤的触觉感受器实现触觉反馈,早期用来向聋人、盲人表达声音、文字和图形信息,逐渐扩展到更为广阔的人机交互领域。与力反馈技术相比,触觉再现技术还远未成熟,目前的触觉再现装置初步用于产生 4 种触觉反馈:

(1) 振动触觉　这类装置利用产生的振动刺激人体的感觉器官,用来表达信息。常见的振动方式是通过电机带动偏心轮来产生振动。另一种方式是

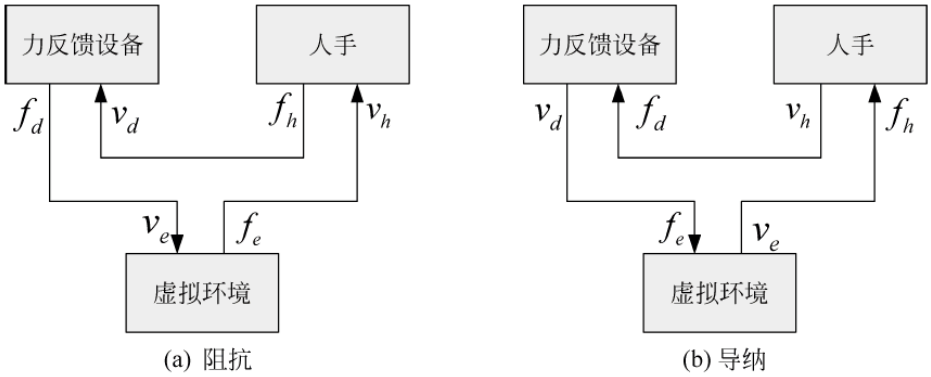


图 1　阻抗及导纳控制

利用压电陶瓷产生高频振动。

（2）形状与压力　这类装置由顶针阵列组成，通过顶针的法向移动来显示形状、轮廓与压力。顶针的驱动方式包括电机、电磁铁、压电晶体和记忆合金。此外，固体顶针也可以被气流或水流代替，利用气动或液压来产生表面形状。

（3）摩擦与纹理　这类装置通过三种方法产生摩擦效果。一是使手指皮肤产生切向变形，从而感受静摩擦。二是通过控制手指与接触表面的相对切向速度模拟滑动摩擦。三是通过改变手指与接触表面的相对摩擦系数，从而模拟纹理。

（4）热反馈　主要使用 Peltier 元件调节手指温度，从而模拟虚拟物体表面的温度。

此外，一些新技术和新原理正在不断应用于触觉再现装置。例如，用电流变液和磁流变液技术来模拟虚拟表面的硬度，采用电极阵列的电触觉（electro-tactile）技术模拟纹理等。

力触觉交互作为人机交互领域的新技术，能够有效促进人与计算机的信息交流。未来力触觉交互技术的发展将追求不断扩展和提升力触觉交互装置的功能和性能，提供更加自然、逼真、方便和可靠的交互方式。随着研究的深入，力触觉交互技术将进一步加强和拓展其在医学、军事、教育及娱乐生活等领域的应用。

参考文献

1. Burdea Grigore C. Force and Touch Feedback for Virtual Reality, John Wiley & Sons, Inc., 1996

2. Biggs S J, Srinivasan M A. Haptic interfaces. In: Stanney K M(Ed.) Handbook of Virtual Environments: Design, Implementation, and Applications. Mahwah, NJ: Lawrence Earlbaum Associates, Inc., London, 2002, 93-116

3. Hayward V, Maclean K E. Do It Yourself Haptics: Part Ⅰ. IEEE Robotics & Automation Magazine, DECEMBER 2007, 14(4): 88-104

（张玉茹　王党校）

lichujue shengcheng

力触觉生成（haptic generating）　借助一些机械和电子设备，向人提供物理刺激如机械力、振动、位移等，从而仿真人与真实环境接触时的力觉与触觉感受的方法和技术。力触觉 haptic 最早来源于希腊语 haptesthai，意思是接触或触摸。它是力觉和触觉的总称。

力触觉生成技术最初用于遥操作机器人控制，它将远地环境中通过力触觉传感器采集的接触信息再现给本地操作者，从而让本地操作者亲身体验远地接触状况，提高对遥操作机器人力位置控制的准确度。随着**虚拟现实**技术的发展，力触觉生成逐渐用于虚拟医疗手术仿真、CAD/CAM、数据**可视化**、远程购物等。在虚拟手术训练中，该技术能帮助医生"触摸"病变组织，提高手术操作准确性。在网上和远程购物中，借助力触觉生成技术，消费者足不出户，就能够触摸感知到想要购买物品的材质属性。

力触觉生成从实现途径上分为直接重构和感官替代两大类。直接重构是直接刺激人的皮肤产生接触感。感官替代是用其他感觉器官替代皮肤接受刺激，间接产生触觉感。力触觉生成可分为力觉生成和触觉生成两个方面。当前力触觉生成方式大多采用直接重构方式。

力觉生成

力觉生成是模拟与环境物体接触时的作用力与变形之间的约束关系。对于接触力的物理建模一般基于力学基本定律如胡克定律等。可分为阻抗式和

导纳式。所谓阻抗式，是根据接触时物体变形量，计算并模拟产生相互作用力；导纳式则是在假设已知物体间施加作用力的大小和状态的情况下，计算并模拟物体的变形/位移。目前阻抗式力觉再现研究较为普遍。这类力觉生成一般经过力觉建模和力觉反馈两个阶段。

力觉建模是建立物体间接触力的数学计算模型。基于物理意义的计算模型总体可分为质点-弹簧模型、有限元模型两大类。质点-弹簧模型是将物体等效为由弹簧和质点组成的系统，质点间相互关系归结为质点间的弹簧作用，按照弹性力学方法分析物体受力和形变之间的关系。有限元模型则是将连续实体离散成有限个单元，建立单元的刚度方程，根据给定的载荷条件和边界位移条件求解总体刚度方程组，从而得到单元所有节点的位移以及应力分布。

质点弹簧模型由于结构简单易用、算法容易实现并且计算复杂度低，已被广泛应用于许多领域。其缺点在于容易受参数的影响、不稳定、仿真效果不理想；有限元模型可伸缩性好，可以很方便地用相同的网格结构实现对不同复杂程度和精度的计算，而且模型的参数易于调节，能够方便地实现材质的各种属性，但是存在的主要问题是涉及大量复杂的计算，计算复杂度高。

值得注意的是，力觉生成和变形计算虽然可以借助于同样的数学模型 $F=f(x,t,v,a)$，但和基于物理意义的变形仿真不同的是，力觉生成与变形计算密不可分。一方面，物体间作用力的计算依赖于人的主动控制（表现为虚拟代理的位移、速度、加速度等），另一方面，物体间作用力又受外力与应力平衡关系模型制约，与物体内部各点的应变有关。因此，力觉生成时，接触力和变形的计算是个反复迭代的过程。要保证力觉生成具有真实感，要求力计算的刷新频率高于 100 Hz，而视觉刷新频率由于变形计算的复杂性，刷新速度一般只有几十赫兹，因此力觉计算要求保证力计算闭环和变形计算闭环的协调性。

经过物理建模计算出作用力后需要装置模拟产生作用力并作用于人。这类装置称为力反馈或者力觉再现装置。力反馈装置不仅能模拟产生物体间的作用力，还能跟踪人手的位置，控制与虚拟环境中物体的交互作用，具有双向信息传递的特点。典型的主动式力反馈装置如图 1 所示，它是由 SensAble 公司研制的具有 6 自由度的 premium1.5 型力反馈装置。



图 1　Phantom 力反馈装置

触觉生成

触觉生成是以机械振动式、气动式、电刺激、热传导等激发方式，模拟人与环境接触时所感知物体的柔顺性、形状、纹理、冷热等触觉特性。触觉再现在模拟触觉感知效果、实现原理、驱动方式上有很大差别。以振动式纹理触觉生成为例，如图 2 所示，它是利用音圈或者压电晶体等产生，振动单元排列成阵列式，通过改变振动单元的振动频率和波形，模拟产生与起伏不平的物体表面接触感。

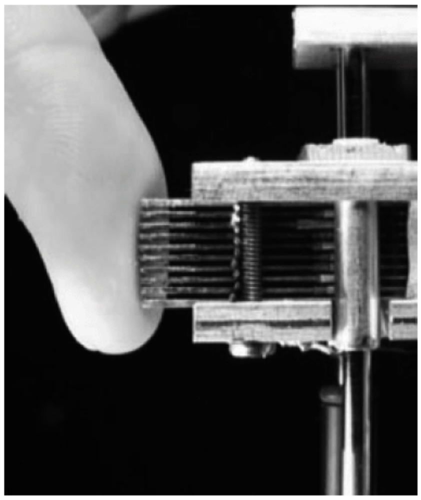


图 2　THMB 触觉显示

未来的力觉触觉生成研究，除了研究实时性和真实感兼备的力触觉计算模型以及多样化的力觉触觉激励方式以外，将集中于研究融合力触觉生成技术。融合的一种途径是将多种触觉装置集成，即力觉和触觉再现装置集成，另一种是多种刺激方式的

融合，例如将热刺激和振动刺激结合，将电刺激、气流刺激相结合等。融合触觉的实现可以以更接近于自然的方式呈现触觉感受，丰富操作者所感知的触觉信息。

参考文献

1. 陆熊，宋爱国. 力/触觉再现中柔性物体可视化物体变形模型研究进展. 计算机辅助设计与图形学学报，2008，20(11)：1389-1395

2. Hayward V. Electro-mechanical transducer suitable for tactile display and article conveyance. US Patent 6,693,516，2004

3. Lévesque V. Braille Display by lateral skin deformation with the STReSS2 tactile transducer. Proc World Haptics Conference 2007，March 22-24，Tsukuba，Japan，2007：115-120

（宋爱国　吴涓）

liti shijue

立体视觉（stereo vision） 采用两个摄像机（双目立体）或多个摄像机同步获取物体图像，计算物体在图像中的差异或视差（disparity），根据视差恢复物体三维信息的方法。

最基本的双目立体几何模型如图 1 所示。它由两个完全相同的摄像机构成。两个图像平面位于一个平面上，两个摄像机坐标轴相互平行，且 x 轴重合。已知一幅图像中的一个投影点，在另一幅图像中寻找对应的投影点，称为对应性求解问题（correspondence）。场景点 P 在左、右图像平面中的投影点分为 p_l 和 p_r。不失一般性，假设坐标系原点与左摄像机光心 C_l 重合。比较相似三角形 PMC_l 和

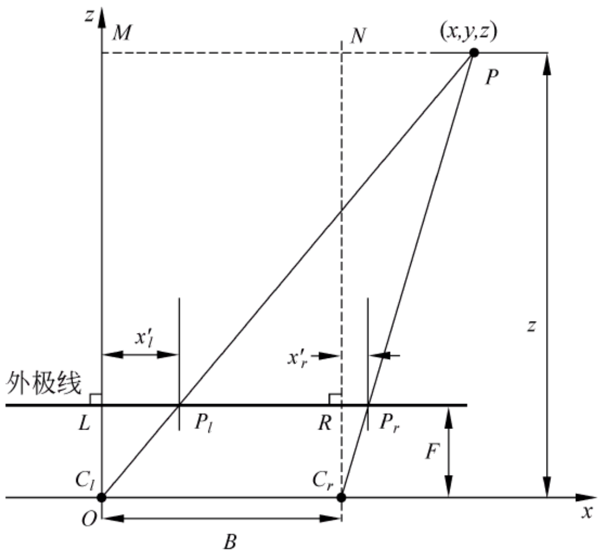


图 1　双面立体视觉几何模型

p_lLC_l，PNC_r 和 p_rRC_r，分别得到

$$\frac{x}{z}=\frac{x'_l}{F},\ \frac{x-B}{z}=\frac{x'_r}{F}$$

合并以上两式，可得

$$z=\frac{BF}{x'_l-x'_r},$$

其中 F 是焦距，B 是基线距离。已知 F 和 B，场景点的深度 z 可以通过求解对应性问题得到视差（$x'_l-x'_r$）来计算。

实际应用中，两个摄像机的光轴不可能完全平行，成像平面不可能共面，因此，场景点 P 在两个图像平面上的投影点不会位于图像中的同一行上，还有光照、变形等问题，使得求解对应性问题极具难度，是**计算机视觉**的经典问题。为了有效求解对应性问题，通常使用外极约束（epipolar constraint）。任意一个场景点和两个摄像机光心构成的平面称为外极平面，外极平面与图像平面的交线称为外极线，经过两个摄像机光心的直线与图像平面的交点称为外极点（epipole）。外极点的属性类似于数学中的极点，但其形成源自图像自身之外的另一幅图像。由此可知，场景点 P 和在两个图像平面上的投影点都位于外极平面上，一个投影点的对应点位于外极线上（图 1 的外极线就是对应的图像行）。外极约束将对应性求解问题的二维（图像）搜索降为一维（外极线）搜索。

立体匹配方法分为基于特征和基于区域两大类方法。基于特征的方法使用图像中具有显著变化的特征（如角点）作为匹配基元，可以得到稀疏的视差图。基于区域的方法使用像素作为匹配基元，对图像中的每个像素分别计算它在另一幅图像中的对应点，产生场景的稠密视差图（dense disparity map）。最常用的匹配测度有规范互相关（NCC）、亮度差分和（SAD 和 SSD）、排序（rank）等方法。

参考文献

Brown M Z，Burschka D，Hager G D. Advances in computational stereo. IEEE Transactions on Pattern Analysis and Machine Intelligence，2003，25(8)：993-1008

（贾云得）

liti xianshi

立体显示（stereo display） 能使人的两眼观察到不同的影像而产生立体感的显示技术，也称为三维显示（3D display）。与传统的二维显示相比，立体显示能够还原出显示内容的深度信息，真实地再现客

观物体,因而能更准确地认识事物的形状和运动形态,使观察者获得身临其境的立体视觉感受。与平面二维显示相比,立体显示的信息量更大,感染力更强,更逼真,具有更强的沉浸感。

目前的立体显示技术分为两大类。即,佩戴立体眼镜的显示技术和裸眼立体显示技术。

佩戴立体眼镜的显示技术 这种立体显示技术是通过佩戴特殊的立体眼镜来实现的。目前立体眼镜实现三维立体效果主要有三种方式:红蓝/红绿眼镜、偏振眼镜和快门眼镜。

(1) 红蓝/红绿眼镜方式 片源是由红、蓝(绿)光染色生成的具有双目视差的立体图像。观察者佩戴的眼镜由红色和蓝(绿)色镜片组成。由于相同颜色的镜片会过滤掉相同颜色的图像,因此,透过红色镜片,眼睛只能观察到蓝(绿)光染色的图像,同样透过蓝(绿)色镜片,眼睛只能观察到红光染色的图像。双眼所看到的图像有差异,从而产生立体视觉。这种方式主要应用于早期的立体电影。其特点是成像原理简单,价格低廉,但三维效果不理想,且色彩还原有偏差。

(2) 偏振眼镜方式 偏振眼镜分为线偏振和圆偏振眼镜两种。线偏振眼镜由偏振角度垂直(正交)的左右镜片组成(通常是0°和90°的偏振片,也可用45°和135°的偏振片搭配)。偏光滤镜或偏光片可以滤除特定角度偏振光以外的所有光。左眼图像经由偏光片进行极化,偏振眼镜的左边镜片使用同一方向的偏光片,因此透过左边镜片左眼能看到左眼图像;同理透过右边镜片右眼能看到右眼图像。由于左右镜片的偏振方向垂直(正交),因此左眼不能看到右眼图像,右眼也不能看到左眼图像。双眼分别看到不同的图像,从而产生立体视觉。圆偏振的原理与线偏振的原理基本相同,所不同的是圆偏振光偏振方向是有规律的旋转着的,它可分为左旋偏振光和右旋偏振光。左、右眼图像分别通过不同旋转方向的偏光片,再经过相应的偏振眼镜镜片,进入左右眼。这种方式在当前的立体电影放映中应用广泛。

(3) 快门眼镜方式 这种方式将左右眼图像按奇偶时序放置,左眼图像在奇数帧,右眼图像在偶数帧。播放左眼图像时,使用专用的眼镜遮住观看者的右眼,播放右眼图像时,遮住观看者的左眼。左右眼只能看到相应的左右眼图像,从而产生立体视觉。将左右眼图像以极快的速度切换,在人眼视觉暂留特性的作用下就合成了连续的画面。目前,用于遮住左右眼的眼镜用的都是液晶板,因此也被称为液晶快门眼镜。这种方式主要应用于立体电视的解决方案中。

裸眼立体显示技术 不戴眼镜的立体显示称为裸眼立体显示。它是立体显示技术的发展方向。在最近几年里,不需配戴眼镜的立体显示器已经从实验室进入市场。

(1) 基于平板显示器的系统 这种立体显示器多数使用光学空分法,把整个屏幕分割成显示像素相互交错的亚屏幕来显示立体图像对(左眼图像和右眼图像),在平板显示器的屏幕前方或后方安装特殊的光学分光器件,使观看空间内形成左右眼独立视区。在对应的独立视区内,左眼只能看到左眼图像,而右眼只能看到右眼图像,经过大脑视觉中枢的立体融合,达到立体显示效果。这种裸眼立体显示器主要采用基于柱透镜方法和视差屏障方法的两种前置分光器件。①柱透镜方法 利用柱透镜实现立体成像源自于视差立体法,即利用人的双眼视差和会聚所构成的深度感实现人意识中的立体感。在液晶屏前面板镶上一块柱透镜板组成立体显示的光学系统,柱透镜板由许多结构参数和性能完全相同的半圆柱透镜紧密排列构成。左眼图像和右眼图像被交错排列在液晶面板的奇数列和偶数列的像素上,所有奇数列的像素构成左眼图像,所有偶数列的像素构成右眼图像。经液晶屏像素调制的光线通过柱透镜的折射,把奇、偶列像素上显示的视差图像投射到人的左、右眼,使人双眼分别看到左眼和右眼图像,就产生了具有视差的立体效果。柱透镜自动分光显示器构造简单,加工和安装比较方便,关键是柱透镜的节距和曲率参数的选取。由于柱透镜与液晶屏是预先固定好的,因此只能用于单一的立体显示,无法兼容平面显示。②视差屏障方法 其成像视觉机理与柱透镜方法相同。视差屏障方法在液晶屏面板前放置一块栅栏式的挡板,人眼穿过一系列的缝隙,左眼只能看到奇数列图像(左眼图像),右眼只能看到偶数列图像(右眼图像),从而形成立体视觉效果。该方法结构简单,根据显示屏尺寸及像素的大小,设计相应的档板间距和狭缝宽度即可,但需要保证显示面、挡板和眼睛的精确位置关系。由于档板位于显示平面的前方,使用电子档板的方法可适用于大部分的平板显示器。

(2) 全息立体显示 全息立体显示是一种真三维立体显示技术。全息技术是利用干涉原理将物体发出的物光波振幅及相位信息以干涉条纹的形式记

录下来，形成全息图。当用相干光源照射全息图时，基于衍射原理重现原始的物光波，从而形成原物体的三维图像。重现的图像立体感强，观看全息立体图像时具有观看真实物体一样的立体感。全息图的每一部分都记录了物体上各点的光信息，故原则上其每一部分都能再现物体的整个图像，通过多次曝光还可以在同一张底片上记录多个不同的图像，且能互不干扰地分别显示。

全息技术是一种非常理想的三维立体显示技术，而且近年来光电技术及器件的发展为数字全息技术的发展和应用提供了良好的基础，也已经成功研发了一些三维全息显示系统，但该技术的研究目前尚未得到本质上的突破，技术难点包括再现影像空间承载问题、再现影像噪声消除问题以及全息真彩色显示问题等。

（3）体三维显示　体三维显示通常是将三维物体分割为点阵或一系列二维图像，再依次扫描，利用人眼的视觉暂留效应形成立体图像。具体实现时，可将一组含有二维图像的光束投影到一个旋转的由特殊材料制造的透明的半球显示器内，经过光学定位系统定位在旋转着的半球面内侧，由于旋转速度很快，打在屏幕上形成的亮点有如悬浮在空中一样，构成了三维图像中的体素。这种立体显示器通常由激光系统、计算机控制系统、旋转显示器系统等子系统组成。

体三维技术能够提供具有真实感的三维图像，在医学上显示人体模型、立体空间中显示物体空间位置、工业上显示复杂机械模型等方面都有其他显示技术无法比拟的优势。这种技术的局限性在于影像中央必须有一个旋转轴，靠近轴心的影像旋转速度较慢，立体影像较不清晰；而且目前人的视点只能游离于三维场景以外来观看，因此很难提供沉浸的效果。

参考文献

王琼华，王爱红．三维立体显示综述．计算机应用，2010，30(3)

（李国宽）

liti xianshi zhuangzhi

立体显示装置（stereo display devices）　用于实现立体显示的设备，一般由光学系统、电子系统、机械系统等在配套软件的控制下运行。立体显示装置是**虚拟现实**系统的硬件基础，涉及光学、电子、机械以及光机电一体化技术等，是集成多学科成果的高科技含量和高技术结晶的复杂仪器系统。

按照工作原理，立体显示装置可以分为以下四种类型：透视型立体显示、视差型立体显示、光学悬浮式立体显示和真三维立体显示。

1．透视型立体显示

透视型立体显示利用阴影和遮挡等多种心理暗示，采用计算机图形学技术对深度信息进行编码，将三维场景以透视形式显示在二维平面上。透视型立体显示只能提供部分的单眼深度暗示，几乎没有运动视差和双眼暗示，仅适用于估计精度要求不高的场合，整体感知能力的提高是以牺牲细节是为代价的。

2．视差型立体显示

视差型立体显示基于双目视差原理，通过计算机合成或者多摄像机从不同角度拍摄获得具有视差的两幅图像，也叫“体视对”，借助光学系统使观察者左右眼只能看到对应视角的二维图像，通过大脑融合成一幅三维图像。根据分离体视对的方式不同，视差型立体显示可以分为眼镜方式和非眼镜方式两种。

通常观看景物时，人眼的调焦和会聚定位在同一个平面上。而在视差型三维显示中，人眼调焦在屏幕上，从而可以清楚地看到视差图，大多数情况下再现的立体图像不在屏幕上，因此人眼的会聚角并没有定位在屏幕上。由于人眼的调焦与会聚不匹配，不符合生理习惯，长期观看容易引起头痛、恶心等反应。但该种三维显示方法具有技术成熟，可实现大景深立体显示的优点，未来的发展方向是超大屏幕立体显示，通过配合立体相机可以实现实时真实场景显示。

3．光学悬浮式立体显示

光学悬浮显示利用光学成像技术改变物体成像位置，广泛应用于幻影仪中。光学悬浮显示设备实现简单，可方便地实现图像的悬空显示，容易将虚拟景物和实际场景结合起来，给人以身临其境的感觉。但严格来说，这种显示不能算是三维显示，因为实际上观察者通过该系统能够看到仅仅是物体某个面的图像，这些图像并不能提供除透视效应以外的深度暗示。

4．真三维立体显示

真三维立体显示通过光学再现或像素填充模拟实际物体各点处的光学特性来再现三维物体，按照工作原理可以分为全息立体显示和体三维立体显示两种。

（1）全息立体显示

全息立体显示利用光的干涉原理，将物体散射

或发射出的特定光波以干涉条纹的形式记录下来，然后利用光的衍射原理，在一定的条件下再现。全息术的特点在于记录了完整的波前信息，能提供一种与观察原物时相同的视觉效果，但其信息量比其他三维图像大了好几个数量级，同时动态全息图空间光调制器的衍射角大小限制了可观察的视场范围。大场景全息三维显示的信息量很大，对空间光调制器、计算机的处理速度、存储容量和传输带宽的要求高，在目前的软、硬件技术条件下难以实现。

（2）体三维立体显示

体三维立体显示按三维显示空间的产生方式，可以分为体积静止型和体积扫掠型两类。

体积静止型的真三维立体显示通过适当的方式来激励或寻址静止体积内的显示介质，并以高于人眼闪烁融合频率的速度进行刷新，以便在每一个指定的空间位置产生体像素，由此构成真实地加载于三维实空间内的图像。由于缺乏合适的显示介质，这种显示方式目前尚无法满足实用的要求。

体积扫掠型的真三维立体显示利用一个二维表面的周期性运动，把预先处理的原始三维信息按时序快速显示在该表面所扫掠的一些空间位置处，再以高于人眼闪烁融合频率的速度进行刷新，由此构成真实加载于三维实空间内的再现图像。这类显示方法按屏幕运动方式又可以分为基于平面屏直线往复运动的真三维立体显示和基于二维屏幕旋转扫掠的真三维立体显示。前者由于机械驱动机构设计难度大无法实现实用，而后者屏幕运动平稳，国内外都已开发出了显示效果较好的显示系统，是目前最有可能实现实用的显示方法。

参考文献

1. Cakmakci O, Rolland J P. Head-worn displays: a review. J of Display Technology, 2006, 2(3): 199-216

2. Jones A, McDowall I, Yamada H, et al. Rendering for an interactive 360 light field display. ACM Trans on Graphics, 2007, 26(3)

3. Cheng D, Wang Y, Hua H, Talha M M. Design of an optical see-through head-mounted display with a low f-number and large field of view using a free-form prism. Applied Optics, 2009, 48(14): 2655-2668

（王涌天　刘越）

licheng

例程(routine)　计算机程序中相对独立、能够完成特定功能的一个代码片段。在整个程序的一次执行中，程序可以从多个不同位置多次调用一个例程。例程执行结束后，程序回到调用点继续向下执行。这里，**程序**一词常指**低级语言**程序。也有人把例程视为**子例程**的同义语。

例程的概念几乎是伴随着计算机程序而产生的。大多数程序都需要将其求解问题的结果输出，这就有了输出例程。它用于启动输出设备，将数据按其规定格式送至输出设备，控制、监督输出操作等。又如，用户常常通过提示指令（命令）的执行顺序，或通过提示执行结果，对计算机程序进行检查，这可以用跟踪例程来完成。除了上面讲到的两个例程外，最常用的服务性例程还有汇编例程、编辑例程、输入例程、故障诊断例程、分类例程等。

一个较大的软件系统往往由若干个例程组成。比如，**操作系统**可由诸如文件管理例程、资源管理例程、作业调度例程、输入输出例程、时钟管理例程、同步出口例程、错误分析出口例程等组成。

例程有可复用例程、可再入例程、递归例程等。

大多数例程一经装入就可执行多次，这便节省了程序设计时间和代码所占有的存储空间，此类例程称为可复用例程。可再入例程是指本例程执行尚未完成，它又可再次进入。一个可再入例程可同时为多个计算机程序使用。在具有多道管理功能的操作系统控制下，语言编译程序应是再入式的。递归例程在程序设计中也是经常需要的。这种例程可直接或间接调用自身。在使用递归例程时，必须将使用过程中尚未用完的状态保存起来。

例程可以是开式的，也可以是闭式的。开式例程在调用处嵌入相应的例程定义；而闭式例程则在调用时直接转至相应的定义，执行后返回。

编译程序目标代码优化的一种技术便是将简短的例程代码嵌入到调用它的地方，借以显著地节省机器执行时间并使存储器占用空间增加不多。RISC 计算机编译程序是采用这种技术的一个范例。

参考文献

Wilkes M V, Wheeler D J, Gill S. Preparation of programs for an electronic digital computer. Addison-Wesley, 1951

（段祥）

lianjie bianji chengxu

连接编辑程序(linkage editor)　把多个分别编译或汇编过的程序段组合成一个大的程序段或程序的程序。有时也称为**连接程序**。

编译程序或**汇编程序**产生的***浮动目标程序***一般由三部分组成:正文,它是目标程序的主要部分,包括指令代码和数据;外部符号(全局符号)表,包括本程序段引用的名字和被其他程序段引用的名字;浮动信息表,包括再定位所需要的有关信息。连接编辑程序扫描外部符号表,寻找所连接的程序段,根据再定位信息表解决外部引用和再定位,最终把多个正文组合成一个待装入的程序。

例如,程序段 A(图 1)和程序段 B(图 2)经连接编辑程序组合后形成程序段 C(图 3)。其中 A 定义名字 D,调用 B;B 引用 D,则 C 中引用 B 定位为引用 l_a+1,对 D 的引用点定位为 l_a+r_d。

参考文献

Beck L L. System software: an introduction to systems programming. 2nd ed. Addison-Wesley, 1990

(张素琴)

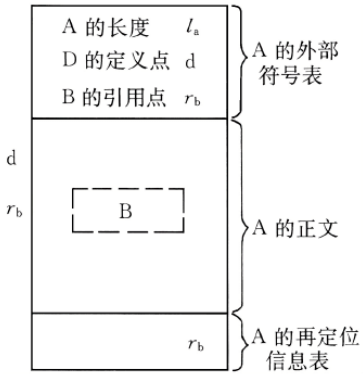


图 1 程序段 A

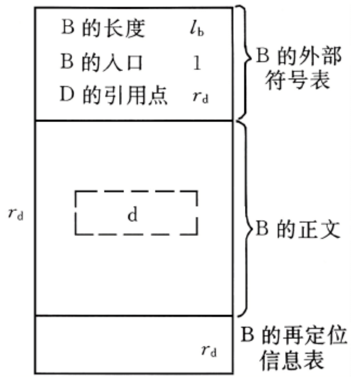


图 2 程序段 B

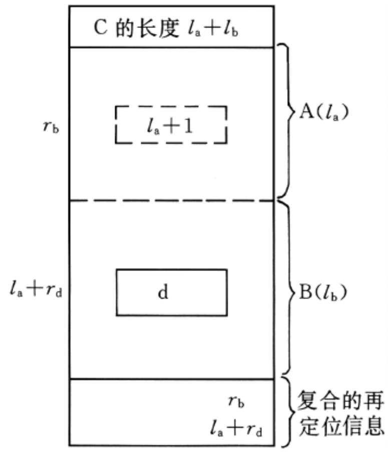


图 3 程序段 C

lianxu shuju baohu

连续数据保护(continuous data protection, CDP) 一种连续捕获和保存数据变化的数据备份技术,将变化后的数据独立于初始数据进行保存,以确保数据可以恢复到过去的任意时间点。连续数据保护系统可以在块、文件或应用级实现,可以为恢复对象提供足够细的恢复粒度,实现几乎无限多的恢复时间点。

连续数据保护有三种实现方式。

(1)基准参考模式 对要保护的数据建立一个初始的映像,然后将数据请求以日志的形式顺序记录。该模式中每次数据请求最多只导致一次对磁盘的实际写操作,因此实现比较简单,带来的额外开销较小。基准参考模式在数据恢复时,需要从最原始的参考数据开始,逐步进行数据恢复,恢复点越靠近当前点,恢复所需要的时间就越长。

(2)复制参考模式 维护的映像是数据的最新状态,该模式克服了基准参考模式中数据读效率低的缺点,适合于读请求较多的环境。复制参考模式通常利用写复制(copy on write)技术,即映像中的原始数据被复写之前先将其复制到日志中。这样记录的日志在恢复点越靠近当前点时所需时间越短,因此,需要较多的系统资源。

(3)合成参考模式 以上两种模式的折中,可以得到较少的资源占用和恢复时间短的效果。但需要复杂的软件管理和数据处理功能,实现起来比较复杂。

连续数据保护实现主要有三类层次。

(1)文件级的连续数据保护 可以捕捉文件系统数据或者元数据的变化事件(例如创建、修改、删除等),并及时将文件的变动进行记录,以便将来实现任意时间点的文件恢复。

(2)块级的连续数据保护 在块设备层实现连续数据保护,屏蔽了与具体文件系统类型的相关性,使得功能适用于各种文件系统。基于块级的连续数据保护功能直接运行在物理的存储设备或逻辑的卷管理器上,甚至也可以运行在数据传输层上。当数据块写入生成数据的存储设备时,连续数据保护系统可以捕获数据的复制并将其存放在另外一个存储设备中。

(3)应用级的连续数据保护 对需要保护的关键应用程序,可以在其中直接嵌入和运行连续数据保护功能。这种实现连续数据保护的方式必须和应用进行深度整合,确保应用数据在连续保护中的一致性。目前基于应用程序的连续数据保护解决方案

大部分是针对成熟的应用开发的。

参考文献

1. Data Management Forum, SNIA. http://www.snia.org/

2. Yang Qing, Xiao Weijun, Ren Jin. TRAP-array: a disk array architecture providing timely recovery to any point-in-time. In: Proc of the 33rd Annual International Symposium on Computer Architecture (ISCA'06), 2006 （曾令仿）

lianxu xitong fangzhen

连续系统仿真（continuous system simulation） 以计算机为工具，对状态随时间连续变化的系统进行仿真的各类活动的总称（参见**计算机仿真**）。连续系统仿真是从时间、数值两个方面对原系统模型进行离散化，并选择合适的数值计算方法来近似积分运算。离散化以后得到的模型称为仿真模型。仿真模型是原连续模型的近似，这样对仿真算法有三个基本要求：

（1）稳定性。仿真模型不改变原模型的稳定性，若原连续系统是稳定的，则离散化后得到的仿真模型也应是稳定的。

（2）准确性。有不同的准确性评价准则，最基本的准则是绝对误差准则与相对误差准则，这取决于仿真要求。

（3）快速性。仿真模型的执行速度。

上述三个方面，其中稳定性是必须保证的，但准确性和快速性两者是相互矛盾的，依赖于仿真要求，往往需要根据仿真要求进行折中。

数值积分法仿真 数值积分法仿真（simulation with numerical integration method）是用数值计算的方法对用常微分方程（和代数方程）描述的系统进行仿真。数值积分法可分为两大类，一类称为单步法，另一类是多步法。

在仿真计算过程中每计算下一时刻的值时，若只需要当前时刻的值，称为单步法。单步法的优点是不仅存储量小，而且可以“自启动”——即已知微分方程的初值时，不必用别的方法启动，可直接用此法进行仿真。另外，仿真步长在整个仿真计算过程中不要求固定，可以根据计算精度的要求动态调整。单步法的缺点是每一步的计算工作量较大，特别是高精度方法需要多次计算微分方程的右端函数值，因而仿真速度较慢。

为提高仿真速度，如果仿真计算时采用前若干步计算得到的数据，称为多步法。多步法仿真的每一步计算不需要多次计算微分方程的右端函数值，计算量比单步法少得多，因而速度快。典型的是**线性多步法仿真**（simulation with linear multi-step method）。它要求有多点初值，这对许多仅具有单点初值的系统来说，仿真计算难以自启动，为此需要用单步法来帮助；另外，它一般还要求在整个仿真过程中仿真步长保持不变。

离散相似法仿真 离散相似法仿真（simulation with discrete similarity method）是一种基于采样定理将连续系统模型进行离散化处理，求得与它等价的离散模型的仿真方法。根据所选用的信号重构与保持器的形式，就可得到不同精度的离散相似法。

由于线性定常连续系统的模型可以用状态方程来表示，也可以用传递函数来表示，因此，与此类连续系统等价的离散模型可以通过两个途径获得：其一是基于状态方程离散化，得到时域离散相似模型；其二是对传递函数作离散化处理得离散传递函数，称为频域离散相似模型。

参考文献

1. Vlach J, Singhal K. 电路分析和设计的计算机方法. 汪蕙，等译. 北京：科学出版社，1992

2. 肖田元，范文慧. 系统仿真导论. 北京：清华大学出版社，2010 （肖田元）

lianji fenxi chuli

联机分析处理（online analytical processing, OLAP） 以**数据仓库**中海量数据为基础的、联机的分析技术和过程。

OLAP 的概念是由 E. F. Codd 于 1993 年提出的。E. F. Codd 是**关系数据库**的创始人，鉴于他的影响，OLAP 技术一经提出便受到了广泛的重视。经过多年的发展，OLAP 技术如今已成为与**联机事务处理**（online transaction processing, OLTP）同样重要的计算密集型应用。

OLAP 支持人们从不同的角度快速灵活地对数据仓库中的海量数据进行联机的复杂查询和多维分析操作，包括切片、切块、旋转、向上综合、向下钻取等。这些操作基于多维数据模型实现。多维数据模型是面向分析的数据模型，是数据分析时用户的数据视图，它给分析人员提供了多种观察数据的视角。

多维数据模型的**数据结构**可以用一个多维方体（CUBE）来表示，即（维 1，维 2，…，维 n，度量值），其中维是人们观察数据的特定角度。例如图 1 中所示

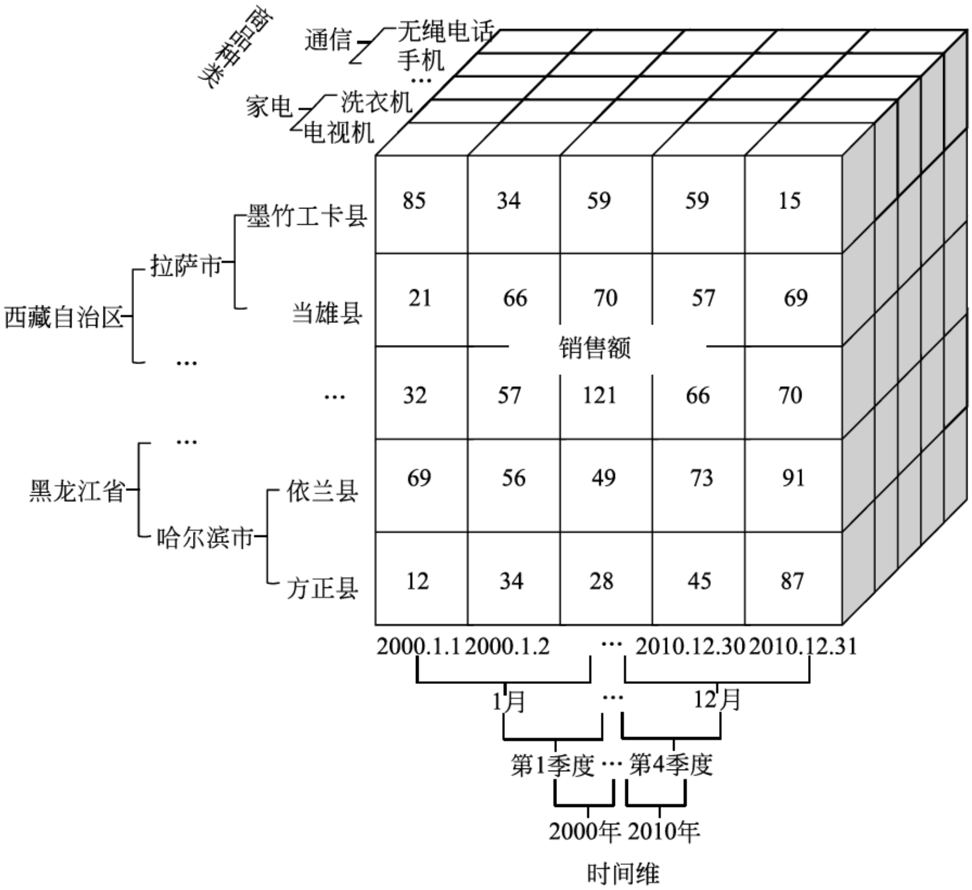


图 1　商品销售的多维数据模型

的商品销售数据是按时间、地区、商品种类,加上变量“销售额”组成的一个多维数组(地区、时间、商品种类、销售额)。它有三个维:地区维、时间维、商品种类维,销售额是度量值。人们可以观察销售额随时间、地区、商品种类及其各种组合的变化情况。维还可能存在细节程度不同的多个描述方面,称为维的层次。例如年、季、月、日等就是时间维的一种层次;同样,县、市、省、国家等构成了地区维的一种层次。

OLAP 服务器软件透明地为上层分析工具软件和用户提供多维数据视图和多维存取接口。分析软件和用户不必关心它的分析数据(即多维数据)到底储存在何处,是如何储存的。OLAP 服务器软件则必须考虑多维数据模型的实现技术,包括如何组织多维数据,如何储存多维数据,多维数据的索引技术,多维查询语言的实现(语言的语法分析、编译、执行和结果表示)和多维查询的优化技术等。

按照多维数据模型的实现方式,可以将 OLAP 服务器软件分为 MOLAP(multi-dimensional OLAP)结构、ROLAP(relational OLAP)结构、HOLAP(hybrid OLAP)结构等。

MOLAP 结构直接以多维立方体来组织数据,以多维数组来储存数据,支持直接对多维数据的各种操作。人们称这种按照多维立方体来组织和储存的数据集合为多维数据库(MDDB)。

ROLAP 结构用关系数据库管理系统(RDBMS)来管理多维数据,用关系(二维)表来组织和储存多维数据,将多维立方体上的操作映射为标准的关系操作。ROLAP 将多维数据划分为两类,一类是事实表,另一类是维表。事实表用来描述和储存多维立方体的度量值及各个维的键值;维表用来描述维信息,包括维的层次及成员类别等。ROLAP 将这些表组织成“星形模式”或“雪片模式”以表示多维数据模型。星形模式通常由一个中心表(事实表)和一组维表组成。星形模式的事实表与所有的维表相连,而每一个维表只与事实表相连。维表与事实表的连接是通过键来体现的。维通常有层次,对维表按层次进一步细化后形成的星形模式的变形称为“雪片模式”。

HOLAP 结构将 ROLAP 和 MOLAP 结合起来,例

如,将细节数据存在**关系数据库**中,而将综合数据存在MOLAP服务器中。HOLAP结构既利用了ROLAP可扩展性好的优点,也利用了MOLAP计算速度快的优点。

E. F. Codd在1993年提出了关于OLAP的12条准则,系统阐述了有关OLAP数据分析的概念、OLAP软件产品的特点和衡量标准,这对OLAP产品的识别及发展都起了重要作用。现在OLAP分析工具及软件产品已在商业数据库领域得到广泛应用。

参考文献

1. Han J W, Kamber M. Data mining: concepts and techniques. 3rd ed. Morgan Kaufmann, 2011

2. 王珊,李翠平,等. 数据仓库与数据分析教程. 北京:高等教育出版社,2012

3. Codd E F, Codd S B, Salley C T. Providing OLAP(Online Analytical Processing) to user-analysts. PC WORLD, 1993

(王珊　陈红　张延松)

lianji shiwu chuli

联机事务处理(online transaction processing, OLTP)　传统**数据库管理系统**的主要应用,指**数据库**应用中基本的、日常的、需要快速响应的事务处理。联机事务处理广泛应用于银行交易、订单处理、电子商务等领域。和联机分析处理相比,联机事务处理主要面向操作人员和低层管理人员。一般而言,联机事务处理所操作的数据库中仅保存数据的最新版本,而不保存随着时间变化的每一版本。联机事务处理中事务执行通常只需要访问较少的数据。联机事务处理的设计和实现强调系统并发性和事务吞吐量。

参考文献

1. Claybrook B. OLTP-online transaction processing systems. New York: Wiley, 1992

2. Garcia-Molina H, Ullman J D, Widom J. 数据库系统全书. 岳丽华,杨冬青,龚育昌,等译. 北京:机械工业出版社,2003

(杨冬青　高军)

lianji shouxie hanzi shibie

联机手写汉字识别(online handwritten Chinese character recognition)　用手或专门的笔在图形输入板上或在手机、计算机等电子设备的液晶屏幕上写字,将字的笔画轨迹、书写顺序和压力大小等信息输入计算机,由计算机自动辨认所写汉字和符号的过程与技术。

计算机从书写的笔画轨迹中提取笔画、笔顺和笔画结构信息,获得**汉字**的笔画结构描述。利用待识汉字的笔画结构描述与标准笔画结构描述进行匹配比较,按其差异大小决定识别结果。由于实时笔画信息的获取,使联机汉字识别的困难程度比脱机识别大为减小(参见**脱机手写汉字识别**)。

若在联机手写汉字识别中利用笔顺信息,则要求书写者的笔顺固定,这样可以极大地简化汉字的结构匹配识别过程;对于无笔顺书写限制的汉字识别,则只能利用书写笔画的相关位置信息进行模式匹配。但是由于不同人书写的笔画轨迹不同,给笔画分割和类型判决带来了困难,使得一般的笔画结构匹配算法遇到较大困难。利用笔画结构的隐马尔可夫模型(HMM)及笔画信息全局统计识别的方法,即在已知笔迹信息条件下的统计识别算法,对无笔顺联机汉字识别取得了重要进展。对于各种草书汉字的识别问题,在获得足够样本的条件下,利用已知笔画信息的全局统计识别算法,是比较容易解决的问题。

目前,当汉字书写比较规则时识别率可达98%。联机手写汉字识别已经广泛推广应用,特别是以嵌入方式在个人数字助理(PDA)和手机中得到了很好的应用。

参考文献

1. 郭平欣,张淞芝. 汉字信息处理技术. 北京:国防工业出版社,1985

2. 张炘中. 汉字识别技术. 北京:清华大学出版社,1992

3. 丁晓青,王言伟,等. 文字识别:原理、方法和实践. 北京:清华大学出版社,2016

(丁晓青)

lianxiang cunchuqi

联想存储器(associative memory)　根据给定内容的特征而不是根据地址进行存取的存储器,它把存储单元所存内容的某一部分作为检索项(关键字项)去检索存储器,并将存储器中与检索项符合的存储单元内容进行读出或写入。又称相联存储器、关联存储器或*按内容寻址存储器*。这种存储器与一般存储器不同之处除了有一般存储器存储信息的功能外,还有对信息进行处理的功能。在计算机系统中,联想存储器主要用于**虚拟存储器**中存放分段表、页表(参见**虚拟存储器**)和快表(参见**转换检测缓冲器**);在**高速缓冲存储器**(cache)中,联想存储器作为存放cache的行地址之用。这是因为,在这

两种应用中,都需要快速查找。此外,利用联想存储器技术还可以构成联想计算机。

访问联想存储器时,将给定的数据特征(或称比较数据)与存储单元中的全部或部分数据进行比较,若符合,则将该存储单元的全部或部分内容按要求读出,或将新的数据写入该存储单元。当有多个存储单元都具有相同的数据特征时,就是多重符合。对于写,可将新的数据都写入这些符合的存储单元。联想存储器框图如图1所示,它由存储单元阵列和一些逻辑电路组成。存储单元阵列中的每个存储单元都有能完成存储、比较、读写、控制等功能的逻辑电路;比较数据寄存器存放要查找的数据,用这个数据的全部或一部分作为数据特征,采用哪一部分可由屏蔽寄存器来决定。比较数据寄存器和屏蔽寄存器是逐位对应的。屏蔽寄存器的某一位为"1",则对应于该位的比较数据寄存器中的那一位作为要查找的特征;反之,如果屏蔽寄存器的某一位是"0",则数据寄存器中的相应的信息位被屏蔽。例如:

比较数据寄存器的内容　1 0 1 1 0 0 1 0

屏蔽寄存器的内容　1 1 1 0 0 0 0 0

则查找的特征是1 0 1 × × × × ×("×"是任意,即"1"或"0")。如果有两个存储单元,它们存放的信息如下:

存储单元1的内容　1 0 0 1 1 1 0 1

存储单元2的内容　1 0 1 1 1 0 1 1

查找的结果是存储单元2的内容符合。

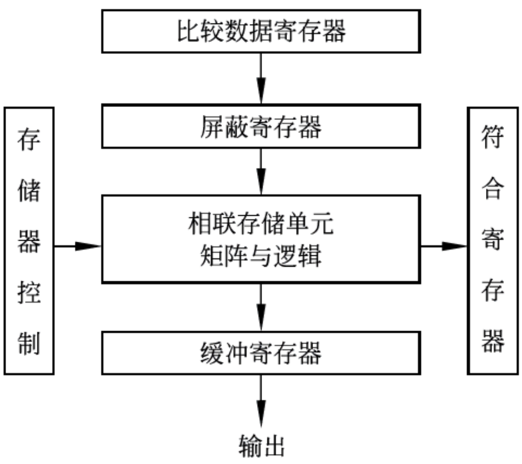


图1　联想存储器框图

符合寄存器中的每一位对应存储单元阵列中的1个存储单元,查找是并行进行的。当某个存储单元的内容符合,则对应的符合寄存器那一位为"1",否则为"0"。当符合寄存器的内容不止是1个"1"时,就发生了多重符合。

联想存储器可实现存储器的并行操作,特别适合于信息检索和更新。它的信息处理功能可以有多种比较操作,如全等、不等、小于、大于等。联想存储器比一般存储器复杂,造价也高,所以一般采用的联想存储器的规模都不大。　(徐炜民)

liangzi chengxu sheji yuyan

量子程序设计语言(quantum programming language)　用以书写量子计算机程序的语言。和经典计算机程序类似,量子计算机程序是量子计算的语言表述,量子计算的基本要素有二,一为计算对象,二为计算规则。在量子计算中,计算对象为量子态,量子态一般为叠加态,为状态向量。计算规则为量子运算。有三类量子运算:即酉运算(亦称么正运算),它是可逆的,用于量子态的改变;测量运算,它是不可逆的,用于输出计算机结果(包括中间结果);张量积运算,用于从分系统的量子态得出复合系统的量子态,或谓用于从分系统的态空间得出复合系统的态空间。

和经典计算类似,量子计算一般包括三个阶段,即输入、计算、输出,输入指制备初态(对应于经典计算的置初值),计算表示量子态的一列改变,输出表示输出结果,通过执行测量运算,使量子态塌缩到一基态,其相应的特征值即对应于计算结果,要注意的是:第一,这种塌缩是真随机的,程序人员无法预知向哪一基态塌缩;第二,不是说,执行测量运算一次,和所塌缩到的基态相应的特征值即为计算结果;原则上说,需重复计算多次,由程序人员适当调整使其具有较高概率幅的特征值方能对应计算结果。

和经典程序设计语言类似,量子程序设计语言亦有低级语言与高级语言之分,低级量子程序设计语言包括量子机器语言与量子汇编语言,前者对应于基本量子运算集,后者是量子机器语言符号化(不仅运算码符号码,地址部分亦符号化)的结果,高级量子程序设计语言则是从程序的易读性与易写性的角度,对低级量子程序设计语言抽象化的结果。当然,量子程序设计语言亦有其他分类方法,如通用量子程序设计语言与专用量子程序设计语言,顺序量子程序设计语言与并行量子程序设计语言等。

鉴于量子不可克隆原理,量子程序设计语言一般不含赋值成分,如需包含赋值成分,则需另附条件。

1982年,Feynman提出了利用量子力学构建计

算机的设想,1985 年,Deutsch 指出,为量子计算机设计程序设计语言为一项有意义的工作,随着 1994 年 Shor 关于大数质因子分解算法与 1996 年 Grover 关于数据库搜索算法等划时代意义工作的出现。1996 年出现了两种量子程序设计语言,即 Knill 的量子伪码(quantum pseudo code)与 Baker 的 Qgol。此后,1998 年,Ömer 设计了 QCL,2001 年 Zuliani 设计了 qGCL,2004 年,Selimger 设计了 QpL,2005 年 Altenkirch 等人设计了 QML,上述工作途径不同,风范各异,堪称量子程序设计语言领域具有代表性的工作。

鉴于一般量子计算问题中既包含量子计算也包含经典计算,故而设想量子计算机的体系结构应由两部分组成,一部分是经典计算机,它负责处理计算问题中的经典计算部分与控制;另一部分为量子设备,负责处理纯量子计算。两部分之间既有分工,又有合作,共同完成量子计算问题中的计算。不过,如果在量子计算问题中,量子计算部分与经典计算部分多次夹插出现,从而导致计算中经典计算机与量子设备的多次切换,这不仅严重影响计算功效,而且会出现难以解决的棘手问题,这也是量子程序设计语言有关成分的设计中难以处理的困难所在。

鉴于现实可用的量子计算机尚未问世,目前各种量子程序设计语言的处理系统只能在经典计算机上模拟实现,从而只能说明可行性,无法说明实用性。

参考文献

1. 徐家福,宋方敏. 量子程序设计语言初探. 中国科学(E 辑,信息科学),2008.06

2. 徐家福,等. 量子程序设计语言 NDQJava. 软件学报,2008,19(1)

(徐家福)

liangzi jisuan

量子计算(quantum computation)　一种基于量子力学基本原理的计算。

与经典计算不同,对于量子计算而言,计算的对象是服从量子力学基本原理的、由量子态表示的量子信息;而计算规则包括封闭物理环境下的酉变换和测量。

量子计算的步骤一般可归结为以下三步:

(1) 初化("入")　这是量子信息"进入"量子计算的过程,指将经典信息通过某种形式的量子编码转化为相应的量子态,即初态制备的过程。

(2) 演化("算")　指代表量子信息的量子态在量子力学基本原理的作用下进行酉变换的演化过程。在此过程中,初化后的信息逐步依照量子算法的步骤转化为隐式表达的计算结果。

(3) 测量("出")　指将一系列酉变换结束后的隐式量子计算结果通过测量(一般为投影测量)转化为经典结果的过程。

以上三个过程在量子计算内部过程中可由图 1 表示。

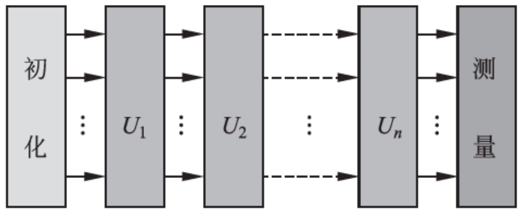


图　1

就发展历史而言,**量子计算机**的概念先于量子计算提出。Feynman 指出,似乎在经典计算机上模拟量子力学系统存在本质困难,并建议在量子力学原理的基础上构造计算机以克服这些困难。1990 年这个思想被具体化并证明在经典计算机上没有已知有效模拟的系统用量子计算机有效模拟是可能的。量子计算机未来的一个主要应用是模拟那些在经典计算机上难以模拟的量子力学系统,这对科学和技术领域有深远意义。

目前看来,与经典计算相比,量子计算具有优越性。这主要体现在以下几个方面:

(1) 量子并行性(quantum parallelism)　对处于叠加态的量子态$|x\rangle$进行U_f所表示的酉变换,这相当于对以量子态$|x\rangle$所代表的变量 x 进行了一次函数作用f,即产生所有关于 x 的对应于函数因变量$f(x)$之量子态$|f(x)\rangle$。量子计算机的这种重要特性称为量子并行性。在经典计算机中,计算所有 x 的$f(x)$需要2^n次计算或需2^n个处理器并行工作,而在量子计算机中,只要 1 次计算(变换)即可完成对所有叠加值的计算。值得说明的是,量子计算虽然可以通过一次操作产生2^n个结果,但只能读出其中的一个结果。由于量子的不可克隆性原理(no-cloning theorem),使得一个未知的量子态不可能被精确复制,这意味着量子计算的结果不可能通过复制而进行保存。

(2) 量子纠缠态(entanglement state)　量子的纠缠态的数学表述为:若一个量子计算系统中多个

量子位的态不能表示成子状态的张量积的形式，则称这多个量子位处于纠缠态，纠缠态首先在 1935 年由 Einstein、Podolsky 和 Rosen 提出。例如 Bell 态：$|\psi\rangle=\frac{1}{\sqrt{2}}|00\rangle+\frac{1}{\sqrt{2}}|11\rangle$ 即为一个典型的纠缠态，其中的两个量子位无法写成张量积形式；当多个量子位处于纠缠态时，对部分量子位的态的测量将影响其他量子位的态的测量，即便这些量子位分处于不同的空间位置。例如测量 Bell 态 $|\psi\rangle$ 时，测量一个量子位的态将使另一个量子位的态必然与之相反。量子纠缠是一种有用的信息资源，在量子隐态传输、量子通信、量子超密编码、量子密钥分配以及在量子计算的加速、量子纠错、防错等方面都起着重要作用。

量子纠缠态使量子计算机具有更大的优越性。相隔很远的两个纠缠的量子态具有瞬时相关性，改变其中的一个状态，则另一个状态立即随之变化，这种关联跨越了空间和时间。与量子纠缠相关的量子隐态传输通信试验已经取得成功。在量子纠缠的情况下，n 个量子位可同时处于2^n个不同状态，这些不同的状态在这些量子位载体上可同时进行计算，产生了量子计算相对于经典计算的指数级加速。

除此之外，量子傅里叶变换（quantum Fourier transform，QFT）、量子隐形传态（quantum teleportation）等都可能是量子计算可能超越经典计算的根源。

许多计算复杂度很高的问题（如 NP 问题）目前尚不能在经典计算机上进行有效计算，由于量子计算的量子并行原理，可以使计算能力比现有经典算法具有指数级的加速。

量子计算必须有高效的量子算法才能发挥其计算优势，目前对量子算法已经有了很多研究。量子并行原理虽然可以仅通过一次变换产生所有 x 的计算结果 $f(x)$，然而测量时只能（高概率的）测得一个结果，而且不能选择所需的结果。量子算法的中心思想就是利用量子态的相干性（coherence），使所需的结果增强，同时使不需要的结果减弱，这样所需的结果在测量时就会以相当高的概率出现。

Shor 算法是量子计算中的一个非常典型而具有标志性意义的算法。整数分解问题（IFP）目前在经典计算机上尚未找到具有多项式时间复杂度的算法，迄今最有效的经典算法之时间复杂度约为 $O(e^{n^{1/3}\lg^{2/3}n})$（注：n 表示待分解整数的位数），如果 n 很大，则计算量是非常惊人的。若 n 为 250 位整数，用 1600 个工作站协同计算估计约需 80 万年。大数质因子分解的困难性是现在广泛应用于电子银行、网络安全等领域的著名的 RSA 公开密钥体系安全性的理论依据。1994 年美国麻省理工学院 Peter W. Shor 提出了大质因子分解的量子算法并证明用此算法若在量子计算机上计算，求一个 n 位（二进制位）大数的两个质因子的时间复杂度约为 $O(n^2\lg n\lg\lg n)$，其仅为多项式。因此，在量子计算机上，整数分解问题可以在多项式时间内解决。这一结果表明利用未来的量子计算机可以在多项式时间内将大数密钥破解，对 RSA 公开密钥系统的安全性提出了挑战。

在快速搜索数据库方面，对 n 个元素的数据表，古典搜索算法的时间复杂度为 $O(n)$。1996 年美国贝尔实验室的 Lov Grover 提出一种基于量子计算的搜索算法，其时间复杂度仅为 $O(n^{1/2})$。这是因为量子计算机将数据库的 N 个被搜索的对象叠加为 Hilbert 空间中的 1 个态，要搜索的态只是其中的一个分量。

量子计算是一种与经典计算迥然不同的新型计算，它将使计算技术进入一种前所未有的新境界。

参考文献

1. Nielsen M A, Chuang I L. Quantum computation and quantum information. Cambridge, UK: Cambridge University Press, 2000

2. Feynman R P. Simulating physics with computers. International J Theor Phys, 1982, 21: 467-488

3. 夏培肃. 量子计算. 计算机研究与发展, 2001, 38(10): 1153-1171

4. 吴楠, 宋方敏. 量子计算与量子计算机. 计算机科学与探索, 2007, 1(1): 1-16

（吴楠）

lingyu gongcheng

领域工程（domain engineering） 针对一组具有相似或相近需求的应用系统建立复用基础设施的**软件工程**。这里领域指一组具有相似或相近软件需求的应用系统所覆盖的功能区域。领域工程覆盖了建立软件的可复用构件的所有活动，是一种系统化的**软件复用**技术。

领域工程对领域中的多个应用系统进行分析，标识这些应用的共同特征和可变特征，对刻画这些特征的对象和操作进行选择和抽象，形成领域分析模型。然后，依据领域分析模型及应用体系结构形成领域中应用共同具有的特定领域软件体系结构

(DSSA)或应用生成过程,并以此为基础识别、开发和组织可复用构件。这样,当开发同一领域中的新应用时,可以根据领域分析模型,确定应用的需求规约,根据领域特定的软件体系结构形成应用的设计,并以此为基础选择可复用构件进行组装,从而形成目标系统。对于一些特定的领域,在工具的有力支持下,甚至可以由新应用的需求规约通过变换直接生成系统。

领域工程分为三个主要阶段,即领域分析、领域设计和领域实现。通常,这三个阶段是顺序进行的。同时,领域工程是反复迭代、逐步精化的过程,在实施领域工程的每个阶段中,都可能返回到以前的步骤,对以前的步骤得到的结果进行修改和完善,再回到当前步骤,在新的基础上进行本阶段的工作。

(1) 领域分析　本阶段的主要目标是获得领域分析模型。在本阶段之前需要进行一些准备工作,例如:确定领域工程的目标,制定领域工程的规划,定义领域的边界,识别信息源等。以此为基础,可以进一步分析领域中应用系统的需求,确定哪些需求是被领域中的应用系统广泛共享的,哪些需求是特定应用系统所独有的,在什么情况下具有这些需求,以及这些需求之间的关系,例如:多选一关系、成组可选关系、依赖关系等。当领域中存在大量系统时,需要选择它们的一个子集作为样本系统。

(2) 领域设计　本阶段的主要目标是获得DSSA。由于领域分析模型中的领域需求具有一定的变动性,DSSA 也要相应地具有变动性。它可以通过表示具有变动性的解决方案等来达到这一点,许多现有的设计模式对于这种变化性都有很好的支持。由于复用基础设施是依据领域分析模型和DSSA 来组织的,因此在本阶段通过获得 DSSA,也就同时形成了复用基础设施的接口和交互规约。

(3) 领域实现　本阶段的主要目标是依据领域分析模型和 DSSA 开发和组织可复用信息。这些可复用信息可能是从现有系统中提取得到,也可能需要通过新的开发得到。本阶段也可以看作复用基础设施的实现阶段。

领域工程是一项需要投入大量人力和资源的活动。但如果实施得有效,其后的应用系统可以基于复用基础设施进行开发,在保证质量的同时,大大降低这些系统的开发成本,从而使软件企业得到较高的回报。

参考文献

1. Guillermo A, Ruben P-D. Domain analysis concepts and research directions. In: Prieto-Diaz R, Arango G, eds., Domain Analysis and Software System Modeling, Los Alamitos, CA: IEEE Computer Society Press, 1991:9-32

2. Gomaa H. An object-oriented domain analysis and modeling method for software reuse. In: Proceedings of the Hawaii International Conference on System Science, Jan. 1992:46-56

(王千祥)

lingyu teding yuyan

领域特定语言(domain-specific language)

用于特定领域问题的编程语言或规约语言。和通用语言相比,通过提供针对特定领域的标记和构造元素,提高表达特定领域问题的直接性和易用性,进而提高解决领域特定问题的功效和质量。

领域特定语言又称为专用语言、任务特定语言、应用特定语言、面向问题的语言等。一般而言,领域特定语言是相对通用语言而言的。但由于“领域”一词的模糊性,一般很难对领域特定语言和通用语言做出严格的区分。例如,一些人认为 COBOL 语言是一种面向商业应用的领域特定语言,另一些人则认为商业应用这种领域过于宽泛,因而不适合将COBOL 作为一种领域特定语言。抛开领域概念的准确定义问题,可以将领域特定语言视为一个持续变化的谱系:谱系的一端是绝对的领域特定语言,另一端则是绝对的通用语言。在这种视角中,COBOL语言可以被认为是在这谱系之中更靠近绝对通用语言端的一种语言。因此,领域特定性不应该被绝对化,而应该被视为一种程度。

从发展历史来看,早期的软件语言通常是为解决特定领域问题而设计的(如 COBOL 语言面向商业应用领域、FORTRAN 语言面向工程计算领域、LISP 语言面向人工智能领域)。但由于这些语言面向的领域范围较广,其已经基本具备了通用语言的能力。在这些早期语言的基础上,产生了更为成熟的通用软件语言(如,PASCAL、C、C ++ 、Java 等)。在这些成熟通用语言不断发展和成熟的同时,面向更为具体的领域问题的领域特定语言也大量产生并得到成功应用(如,HTML, CSS, SQL 等)。在当前阶段,通用语言和领域特定语言成为两种相对独立且相互补充的重要方面:其中,软件复用成为领域特定语言再次受到重视的重要驱动因素。

从软件复用的角度来看,领域特定语言体现了一种生成式的软件复用途径。观察软件应用的本

质,可以发现其中通常包含三类成分:通用基本成分、领域共性成分以及应用专用成分。依据复用方式的不同,软件复用可以分为组装式复用和生成式复用:前者指复用已有的软件构件,通过构件集成(组装)得到新的软件应用;后者指复用已有的软件开发过程,通过将使用特定领域语言编写的程序输入可复用的应用生成程序来自动或半自动地生成新的软件应用。但在实践中,生成式复用和组装式复用两者并非截然对立,而具有密切联系:使用领域特定语言编写的一段程序在经过应用生成程序的处理并进行必要封装之后可以作为一个构件用于组装式软件复用中;反之,一个应用生成程序也可能将领域特定语言程序变换为对一组可复用构件的组装和调用。

从功能上来看,任何一种通用语言,在面向特定领域的函数库的支持下,都可以被视为一种领域特定语言。基于这种观点,领域特定语言似乎并没有独立存在的必要性。但是,通用语言往往只能提供有限的自定义/扩展机制;因此,通过为通用语言附加特定领域函数库而得到的领域特定语言,可能无法具备充分的领域特定性。另一方面,虽然无法提供充分的领域特定性,但特定领域函数库仍然是领域特定语言的有力竞争者:脱离通用语言的支持,开发一种具有更好领域特定性的新语言,具有较高的风险性,而且可能需要付出远大于开发一个特定领域函数库的成本;大量成熟的面向特定领域的构件框架的出现,则能进一步降低通过“通用语言 + 特定领域函数库”提高领域特定性的风险和成本。

参考文献

1. 杨芙清,梅宏,李克勤. 软件复用与软件构件技术. 电子学报, 1999, 27(2): 68-75

2. van Deursen A, Klint P, Visser J. Domain-specific languages: an annotated bibliography. ACM SIGPLAN Notices, 2000, 35(6): 26-36

3. Mernik M, Heering J, Sloane A M. When and how to develop domain-specific languages. ACM Computing Surveys, 2005, 37(4): 316-344

(张伟 梅宏)

liulanqi

浏览器(browser) 让计算机用户在**万维网**中查找和阅读信息资源的软件,是万维网最通用的客户端软件。信息资源可以是存储在服务器中的各类文件,包括网页、图形、声音、视频、文件以及应用程序等,用 URL 地址进行标识。

浏览器根据用户请求,获取**万维网**(WWW)服务器上用**超文本置标语言**(HTML)编写的信息资源文件,解释这个文件,并将文件内容以用户环境所许可的效果最大限度地显示出来。当用户选择一个超文本链接时,浏览器通过超文本链接相连的统一资源定位地址(URL)来请求获取其他信息资源文件,等待服务器发送文件,然后处理这个文档,并在客户端显示出来。

第一个基于文本的浏览器诞生于 1991 年。1993 年发布的 Mosaic 浏览器使得万维网获得快速发展。这个浏览器通过鼠标单击方式进行操作,支持图形界面,对网页中的 HTML 标记进行解释并按规定的格式进行显示。1994 年,网景导航者浏览器(Netscape Navigator)在其发布不久即成为具有统治地位的浏览器。一年之后微软公司发布的 IE 浏览器成为应用最为广泛的浏览器。

当前主流的浏览器包括 IE、Firefox、Chrome、Safari、Opera 等,适用于不同的计算机、操作系统及用户终端。

浏览器具有兼容性问题。由于不同浏览器采用的内核不同在解释 HTML 语言时使用的方法不同以及显示器的环境不同等因素,导致浏览器在客户端所显示的网页效果并不相同,甚至出现位置混乱、错位等现象。目前还没有解决兼容性问题的统一工具,最普遍的解决办法是使用各种浏览器进行测试,通过调整样式控制以及脚本控制,调整网页在各种浏览器中的显示效果。

目前,浏览器主要向高速、稳定、安全、开放、多元化、多核化、个性化、支持移动终端特性等方向发展。

参考文献

1. 胡道元. 计算机网络(高级). 北京: 清华大学出版社, 1999

2. Tanenbaum A S. 计算机网络. 4 版. 北京: 清华大学出版社, 2004

(胡道元 张蓓)

liulanqi-wanweiwang-shujuku moshi

浏览器-万维网-数据库模式(browser-Web-database mode) 一种三层客户-服务器模式。又称 BWD 模式(参见**客户-服务器计算**)。在 BWD 模式中,浏览器是客户,数据库端是服务器,而万维网(Web)扮演着应用服务器的角色。

这种三层计算模式的最大优点是提供统一的用户界面,可以支持各种计算机、各种操作系统、各种

数据库管理系统以及各种用户界面。

这种三层计算模式能提供功能性数据库服务器管理，可以优化数据库服务器的存取管理，并且这种优化与具体的数据库管理语言无关。作为中间层的Web服务器完成过程管理功能，为客户提供与数据库服务器无关的统一界面。由于应用逻辑都集中在中间层上，使得对这些逻辑的修改和管理的复杂性减少。而对于两层的客户-服务器计算模式，对应用逻辑的任何修改都导致对每个客户应用的修改。

BWD模式的另一个优点是对事务的可靠控制。由中间层Web管理分布式数据库的事务，通过名字而不是通过位置来访问资源，因此提供了更大的伸缩性和可扩展性。

在BWD模式中文本数据库存储现存的文本，且能转换到HTML格式，并在Web浏览器上显示。通用网关接口(CGI)是数据库服务与HTML文件之间的接口程序，负责处理HTML文件运行在数据库服务器中的非HTML程序之间的数据交换。当用户从浏览器输入查询信息或传送数据后，便激活一个CGI程序。该CGI程序又可调用操作系统下的其他程序，完成用户的查询任务，并将查询结果传给CGI，通过CGI传给Web服务器，其交互流程如图1所示。

参考文献

胡道元. 计算机局域网. 4版. 北京：清华大学出版社，2010

（王勇　相士俊）



图1　BWD模式中CGI与服务器交互的流程

liu celiang

流测量(flow measurement)　指以IP流为单位对网络流量进行观测。所谓IP流，是指符合特定流规范(specification)和超时(timeout)约束的一系列IP报文的集合。IP流可以分为双向流和单向流，双向流是将流规范所规定的对应源宿端点的所有报文均归入该IP流，而单向流对应地将报文按照不同传输方向将双向流区分为两个IP流。

流规范通常通过IP报头中的各个字段来定义，目前使用频率较高的主要有由源宿IP地址、源宿端口和协议类型等五个字段确定的五元组以及源宿IP地址对、宿IP地址等，最常用的是五元组流规范。

流测量功能通常是作为一种流量报告机制实现在网络设备(路由器/三层交换机)中，输出的测量结果为流记录，其中不仅包含了作为标识用的流规范信息，也包含了关于这个IP流的统计信息，包括构成该IP流的IP报文数、字节数以及某些重要的特征信息，例如采集到该IP流的设备端口标识等。受网络设备存储资源的限制，通常无法在这些设备内存中长时间维持过多的流记录，因此需要使用超时约束将超过一定时间不活动的IP流定义为已终结，并将其清除出流测量缓存，以限制并存的流记录数量。超时策略的设定对流测量的精度和测量系统资源的利用状况有较大影响，一般常采用64 s作为默认配置。但是越来越多的研究表明，随着网络信道容量的提升，较短的超时约束(例如16 s)更为合理。流测量功能可以完全使用软件实现，例如Cisco开发的Netflow；也可以使用专用的设备芯片来实现，例如由InMon、HP和Foundry Networks联合开发的sFlow。

由于流测量可以提供SNMP协议所不能提供的网络流量细节，支持对网络流量和网络服务进行行为分析和异常检测，因此成了网络管理系统的一个重要的数据源。网络管理系统可以使用UDP作为传输协议，从网络设备中读取流测量结果。为支持流记录格式的规范化和标准化，互联网工程任务组成立专门的IPFIX(IP Flow Information Export)工作组，以Cisco的NetFlow v9为基础，制定了用于描述IP流的IPFIX规范和相应的用于从网络设备获取流记录的IPFIX协议。基于这个规范和协议，网络管理系统可以根据自身的需求定义特定的流记录模板(完整流记录格式的子集)，并要求被管的网络设备按此模板进行流测量，具有很好的灵活性和可扩展性。

IP数据流被定义为在网络中传输的具有共同属性的一系列IP报文。共同属性是指IP报文的某些字段内容相同或报文的某些特征一致，例如具有相同源/目的地址、源/目的端口、协议类型、服务类型(type of service)的报文可被归为同一条流，或具有相同MPLS标签的所有报文以及具有相同输入/

输出的逻辑接口、下一跳等的报文也可被定义为同一条流。这样定义的流不一定与应用层的数据流一致。

按照对 IP 数据流的处理过程,可分为流导出设备和流采集设备。流导出设备上有一个或多个观测点,观测点的集合构成观测域。例如一台路由器上有多个接口(物理接口或逻辑接口),每个被关注的接口构成观测点,整体构成一个观测域。流导出设备上的计量进程负责对经过观测域的报文进行筛选分类等操作,将满足相同属性条件的报文汇总,形成流记录(flow record),流记录不再活跃或满足失效条件后,该条目及相关的统计信息被导出。流记录信息按一定的周期导出,流采集设备被动的接收导出的流信息。

导出流记录依照 IP 流信息导出协议进行。该协议采用模板来定义输出的内容,模板定义了如何解析数据集中的内容。为了方便对内容进行解析和减少输出的长度,IPFIX 还定义了信息模型对常见的需要输出的内容进行了结构化描述。只要指定预设的信息模型编号,即确定了相应的字段长度、特征描述、制约关系等。IP 流信息导出的传输层协议可使用 TCP 或 UDP,但协议本身建议使用流控制传输协议(Stream Control Transaction Protocol, SCTP)作为其传输层协议。SCTP 提供的服务类似于 TCP,同时又具备 UDP 的一些优点,是一种可靠、高效、有序的数据传输协议。IP 数据流导出还提供了对流数据进行匿名处理、SNMP 管理以及导出双向流等特性的支持。

参考文献

1. 程光,龚俭. 互联网流测量. 南京:东南大学出版社, 2008

2. http://datatracker.ietf.org/wg/ipfix/charter/

3. 互联网级知识系统. The System for Internet-Level Knowledge(SiLK). http://tools.netsa.cert.org/silk/index.html

(龚俭 周昌令)

liu meiti

流媒体(streaming media) 以流的形式在网络(互联网、无线移动网络等)上进行数字媒体(主要指音频和视频)传输的技术。它将视频、音频之类的连续媒体经压缩编码、数据打包后按照一定的时间间隔要求连续地发送给接收方,接收方在后续数据不断到达的同时对接收到的数据进行重组、解码和播放。

过去,多媒体文件(节目)需要从服务器上下载到终端设备后才能播放,这限制了人们在计算机和互联网上使用多媒体数据进行实时的交流。流媒体技术能够较好地解决这个问题。它的主要特点是边下载边欣赏,从而使用户能够不间断地在线欣赏多媒体节目。

流媒体技术所研究的主要内容如下。

流媒体编解码技术:典型的流媒体编解码技术有用于 64kb/s 视频传输的 H.261,面向 1.5 Mb/s 数字视频音频传输和存储的 MPEG-1,面向高品质数字视频音频传输和存储的 MPEG-2,面向交互应用和网络传输的 MPEG-4、H.264 以及适于低码率视频编码的 H.263。最新的发展趋势是可扩展性编码,如细粒度可扩展(FGS)编解码技术和 H.264/SVC(scalable video coding)等。

流媒体存储和调度策略:网络带宽和视频服务器的输入、输出往往是制约流媒体服务性能的瓶颈。在大规模点播电视(VOD)系统中,用户对媒体数据的点播往往集中于少数热门节目,流媒体存储和调度策略的关键是合并用户服务,共享服务器和网络带宽资源。

流媒体的传输与控制:解决媒体流在两个端系统间传输的相关问题,包括媒体流拥塞控制策略、差错控制策略、速率调节策略等,其目标是提高流媒体应用的服务质量(QoS)。

多媒体代理服务器及内容替换机制:多媒体代理服务器将一些访问频繁的多媒体数据存储在内存或硬盘中,当用户通过多媒体代理服务器访问这些数据时,多媒体代理服务器无须访问远程 Internet,而是通过本地缓存为用户提供服务,有效降低了远程服务器的访问负载,节约了从远程服务器到代理服务器间的网络资源消耗,并能有效降低用户的启动延迟,提高用户接收到的媒体质量。近几年,基于 P2P 和基于云计算、云存储的流媒体技术也得到了广泛的应用。

流媒体的典型应用包括点播电视、**视频会议**、远程教学、**数字图书馆**等。

参考文献

钟玉琢,向哲,沈洪. 流媒体和视频服务器. 北京:清华大学出版社,2003

(钟玉琢 孙立峰)

luyou jizhi

路由机制(routing mechanism) 决定在网络中数据经由何种路径从一个网络节点传送到另一个

节点的机制。通常是指数据从其源节点传送到其目的节点的机制。路由问题是通信网络中的基本问题。互连网络的发明和使用以及并行和分布式计算机的建造和使用为路由技术的研究提供了新的需求和推动力。以下主要介绍并行与分布式计算机系统中的路由机制。

与互连网络和其他通信系统中类似,电路交换和分组交换是并行与分布式计算机系统中计算节点间的通信或者存储节点与处理节点间通信的两种信息交换模式。电路交换模式早已用在传统的通信网络中(如电话网络中)。分组交换方式发明于20世纪60年代。在电路交换模式中,通信的源节点和目的节点之间在传送数据之前需建立一条电路,而这条电路需在数据传送过程中一直保持建立。分组交换则不同,这种交换方式中需要传送的数据会分成若干分组或称为数据包,结点之间的连接仅需在一个数据包传送时维持,同一次数据传送所分解成的各个数据包可能经由不同的链路传送,甚至可能到达其目的地的先后顺序与其发送的顺序也不同(从而需要在其目的结点把各个数据包重新进行组装)。交换方式对于路由机制和算法的设计有重要的影响。

路由算法就是决定数据传送所要经由的路径的算法。在并行计算机或分布式计算机系统中,可以有三种机制实现路由的设定。第一种是采用简单的算术或逻辑的计算获得路由,即当前的路由器按照已知的网络拓扑结构,按照地址关系计算出路由。在并行计算机和分布式计算机系统中,其互连网络通常具有规则的拓扑结构,如树形网络、环形网络、网格形网络、超立方体网络,或者各种动态网络(如蝶形网络、Ω-网络等),其路由可以由源地址和目的地址快速计算出来。比如在网格网络中,只要知道源地址和目的地址,即可快速计算出经过几次东(西)方向和几次南(北)方向的传输即可把数据传送到目的地;在超立方体网络中只要把源地址和目的地址做按位异或运算即可得出其按照各个维度的传送路径。第二种被称为基于源结点的路由机制。在这种机制中,数据包从源结点到目的结点的路由(即其传送路径上要经过的中间结点号码),在数据包注入到网络中时就被源结点计算出来,并且被表示在数据包的头部,其后数据包便按照此路径逐一经过中间结点而传送到目的结点。第三种机制就是路由表。此种方式中,每个结点均设置一个路由表,路由表表达了数据包的目的地址和在当前路由器中的输出通道的关系。每当一个数据包需要传送时,可以根据此种关系在路由表中查到其下一跳的地址和通道号。这种方式通常应用在分布式计算机系统中。无论哪一种机制中,路由均由相应的路由算法计算得出。

路由算法可以是确定性的算法或者是适应性算法。确定性的路由仅仅根据源结点和目的结点的关系确定传送路径,而与该路径上可能的现实通信情况无关。例如,在网格网络中的X(东、西)、Y(南、北)方向顺序确定的算法,在超立方体网络中的*e*-cube算法或其他*k*-ary *n*-Cube的维度顺序的路由算法。适应性的路由算法不但考虑源地址和目的地址的关系,而且要考虑网络中通信的现实状况以便在数据传输过程中动态调整传输的路径。因此,适应性路由算法要能计算多条可行的路径并且选择其中"最好的"路径而作为实际传送数据的路径。

路由算法设计中的一个重要问题是避免死锁。当两个或多个数据传输均需要获得某些资源而其中任何一个需求均得不到满足时,就产生死锁。与其他领域中避免死锁的研究类似,在避免死锁的路由的研究中采用资源(通道)使用的依赖关系图是一种有效的方法。路由算法的设计中还要考虑尽量避免路由的冲突,包括传送结点上的冲突或者传送链路上的冲突,这可以减少通信链路上的拥塞,减少通信延迟并提高通信系统的吞吐率。考虑到网络中可能发生结点故障或者链路故障,在路由算法的设计中还要考虑容错的路由算法,以保障网络中出现故障时也可以进行通信。随着计算系统规模的扩大,能量消耗越来越称为信息系统设计的重要问题。尽量减少能耗的路由和综合权衡能耗和其他指标(例如通信延迟、系统吞吐率)的路由是路由算法设计的一个重要考虑因素。当前,多核和众核计算机体系结构是高性能计算机体系结构发展的重要趋势,而片上网络(network on chip,NoC)已经成为多核和众核系统设计中的重要技术。虽然片上网络和非片上实现的网络以及他们的路由机制和算法的设计原理上相似,但其物理尺度、时间尺度和资源限制均对于片上网络及其路由机制和算法的设计提出新的要求。路由机制和算法的设计要综合考虑系统环境和应用特征的各种要求,系统权衡,做出优化的设计。

参考文献

1. Culler D E, Singh J P, Gupta A. Parallel computer architecture: a hardware/software approach. 2nd ed. Morgan Kaufmann, 1999

2. Leighton F T. Introduction to parallel algorithms and architectures: arrays, trees, and hypercubes. Morgan Kaufmann, 1994

3. Patterson D A, Hennessy J L. Computer architecture: a quantitative approach. 5th ed. Morgan Kaufmann, 2011

（刘志勇）

luyouqi

路由器(router)　工作于**开放系统互连参考模型**(OSI/RM)的第三层——网络层的一种**网络互联设备**。它采用某种路由算法,为在网络上传送的数据包从若干条路由中选择一条到达目的地的路径并进行转发。

使用路由器可以实现具有相同或不同类型的网络的互联。以前曾把完成这一功能的设备叫做**网关**,现在把它称作路由器以强调它在 OSI/RM 第三层上的网络互联功能。

路由器要完成的主要功能是路由选择。**路由选择**是通过路由器中的路由表来完成的。路由表中的路由表项一般可以分为静态和动态两类。静态路由表项一般由系统管理员来手工设定,对于路由器的负荷开销比较小,但是不能根据外部网络的变化而自动调整。动态路由表项对路由器有一定的开销,但是可以根据外部网络的变化根据路由协议自动调整,不需要系统管理员的人为干预。

按照支持的协议,路由器可分成单协议路由器和多协议路由器两大类。单协议路由器用于具有相同网络层协议的网络互联;多协议路由器可以用于具有不同网络层协议的网络互联,如 IP、IPX、AppleTalk、Banyan VINES 等。

按照所处的网络位置,路由器可分为接入路由器和核心路由器两大类。接入路由器位于网络外围(边缘),将家庭或 ISP 内的小型企业客户连接到 Internet。核心路由器位于网络核心,通常用于连接不同的网络,起到一个数据转发的桥梁作用,一般具有较大的吞吐量。

按照硬件体系结构,路由器可以分为第一代单总线单 CPU 结构路由器、第二代单总线主从 CPU 结构路由器、第三代单总线对称式多 CPU 结构路由器、第四代多总线多 CPU 结构路由器、第五代交叉开关/交换式体系结构路由器和第六代共享并行处理器和全光空分交换结构路由器等多类。

参考文献

1. Tanenbaum AS. Computer networks. 4th ed. Englewood Cliffs, NJ: Prentice Hall, 2004

2. 雎丹,马海军,纪多辙,向方,等. 网络设备基础教程与实验指导. 北京:清华大学出版社,2007

（毕军）

lunyu lilun

论域理论(domain theory)　由 D. Scott 于 20 世纪 60 年代创建的**指称语义**的数学基础。简称域论。

指称语义的特点就是把每个语言成分映射为一个数学对象,然后用此数学对象上的运算来表达该语言成分的语义。但是,这个数学对象往往是递归定义的。例如,考查下列用 BNF 写的表达式定义(参见**巴克斯范式**):

$EXP ::= NUM \mid AEXP \mid CEXP$

$NUM ::= DIGIT \mid NUM\ DIGIT$

$AEXP ::= EXP + EXP \mid EXP - EXP \mid EXP * EXP$

$CEXP ::= \underline{\text{if}}\ COND\ \underline{\text{then}}\ EXP\ \underline{\text{else}}\ EXP\ \underline{\text{fi}}$

$COND ::= EXP = EXP \mid EXP > EXP \mid EXP < EXP$

$DIGIT ::= 0|1|2|3|4|5|6|7|8|9$

与之相应的语义域方程是:

$$Exp = Num + [Exp \times Exp \to Exp] + [Cond \to Exp]$$

只有解出这个语义域方程,才能得到上述表达式的语义。这里有两个关键问题。第一,是否存在一种数学对象,它是该域方程的一个解?如果存在,能否把它构造出来?第二,是否存在一种有效的方法,可以定义这种数学对象上的可计算函数,并且把描述程序设计语言的指称语义所需的函数均包括在内?

D. Scott 对这两个问题都给出了回答。存在着一种叫连续格的数学对象,它可以作为递归域方程的解;存在着一类称作连续函数的函数类,它在连续格上是可计算的。并且这个函数类足够大,可以满足描述一般程序设计语言指称语义的需要。

后来,D. Scott 把连续格上的连续函数理论进一步发展为论域上的连续函数理论。其概要如下。

具有偏序关系$\sqsubseteq$的集合 D 称为偏序集,其中$\sqsubseteq$是 D 上的自反、反对称和传递关系。D 的子集 M 称为是有向的,若对每个有限集 $M' \subseteq M$,存在 M' 的上界 $x \in M$。偏序集 D 称为是完备的,若它的每个有向子集 $M \subseteq D$ 都有一个最小上界 $\sqcup M$,且 D 有一个最小元素 $\perp$。完备的偏序集称为 cpo。D 的元素 x 称为是紧致的,如果对 D 的任意有向子集 M,使得 $x \sqsubseteq \sqcup M$ 者,必存在 M 中的元素 y,使得 $x \sqsubseteq y$。令

$K(D)$表示 D 中全体紧致元素的集合,则 cpo D 称为是代数的,如果对每个 $x\in D$,集合 $M=\{x_0\in K(D)\mid x_0\sqsubseteq x\}$是有向的,且 $\sqcup M=x$。D 称为是一个论域,若 D 是代数的且 $K(D)$是可数的。

设 D_1 和 D_2 都是 cpo,f: $D_1\to D_2$ 是映 D_1 入 D_2 的函数,若对 $x\sqsubseteq y$ 必有 $f(x)\sqsubseteq f(y)$,则称 f 是单调的。若 f 是单调的,且对 D 任意的有向子集 M 有 $f(\sqcup M)=\sqcup f(M)$,则 f 称为是连续的。如果还有 $f(\perp)=\perp$,则称 f 是严格连续的,用 f: $D_1 \multimap D_2$ 表示。

由于在指称语义中不仅使用一般函数,而且使用高阶泛函,因此上述的论域及连续函数的条件还太一般。例如,若 D_1,D_2 都是论域,f: $D_1\to D_2$ 是连续函数,则所有这些 f 虽然构成一个 cpo,但尚不一定构成一个论域。为此需要对论域作进一步的限制。D. Scott 给了一个一般的条件:可有效表示的论域,并证明了:若 D_1 和 D_2 都是可有效表示的论域,则全体连续泛函 f: $D_1\to D_2$ 也构成一个可有效表示的论域。语义域上的操作,如乘积、求和、λ 抽象等,都属于这类连续泛函。利用最小不动点方法即可确定语义域上连续泛函的解。

如果程序中有不确定性,则程序对同一组数据(同一个初始状态)的计算结果将有多种可能性(不同的后续状态)。用映射来刻画这个程序的执行时,它的映像就不是由(通常)元素构成的集合,而是由集合构成的集合。这表明,不能使用通常的平坦论域,而要使用平坦论域的幂域。为了描述这类程序的指称语义,需要把上述论域理论推广为幂域理论。

建立幂域理论的关键是把普通论域中的偏序关系推广到幂域中去。在传统的幂集中,$A\sqsubseteq B$ 当且仅当集合 A 包含在集合 B 中。对描述指称语义来说,仅有这种规定是不够的。两个互不包含的集合之间也可以有偏序关系。对此已引进了三种不同的偏序关系。

1. 霍尔偏序:$A\sqsubseteq_E B$ 当且仅当 $\forall x\in A$, $\exists y\in B$,使 $x\sqsubseteq y$

2. 史密斯偏序:$A\sqsubseteq_M B$ 当且仅当 $\forall y\in B$, $\exists x\in A$,使 $x\sqsubseteq y$

3. 普洛特金偏序:$A\sqsubseteq_{EM} B$ 当且仅当 $A\sqsubseteq_E B$ 且 $A\sqsubseteq_M B$

基于不同的偏序关系,将得到不同的幂域理论,它们各有其适用范围。例如,为了证明程序的部分正确性,可以使用霍尔偏序;为了证明程序的健壮正确性(只要规约正确,程序一定正确),可以使用史密斯偏序;为了证明程序的完全正确性,可以使用普洛特金偏序。

参考文献

1. 陆汝钤. 计算系统的形式语义. 北京:清华大学出版社,2017

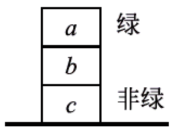
2. Van Leeuwen J. Handbook of theoretical computer science, Volume B, formal models and semantics. Cambridge, MA: MIT Press, 1994

(陆汝钤)

luoji biaoshi

逻辑表示(logic representation) 使用特定的逻辑公式表示知识的一种基本的**知识表示**方法。

对于逻辑方法的研究,最早可追溯到 2000 多年以前亚里士多德的《工具论》。现在的数理逻辑主要起源于 17 世纪后期,其中,G. W. Leibniz 首先明确提出了数理逻辑的指导思想。此后,从 18 世纪到 20 世纪 30 年代,有许多学者开展了系统而深入的研究,把算术、代数、初等数论、集合论等用于逻辑研究,奠定了数理逻辑的基础,其中,弗雷格建立的一阶逻辑对后来的人工智能和计算机科学产生了深远影响。一阶逻辑及其子集命题逻辑是人工智能领域最早采用的知识表示方法之一,从 20 世纪 50 年代后期开始就用于研究开发诸如自动定理证明或问题解答等人工智能系统,其中较著名的有:命题逻辑自动定理证明系统 Logic Theorist;采用一阶逻辑表示知识,运用归结原理的通用问题应答系统 QA3 (C. C. Green 等人),采用命题逻辑编码的规划器 SatPlan(曾获多次国际规划竞赛冠军)等。命题逻辑和一阶逻辑皆属于形式逻辑,统称标准逻辑。这类可用于知识表示的符号化形式逻辑系统,是由一组严格定义的符合语法规则的符号或符号串(即合法语句)以及一些严格定义的、对这些合法语句做出解释或进行操作的形式方法(即形式语义和推理规则)所组成。其中一阶逻辑是命题逻辑的扩充,比命题逻辑具有更强的表达能力,所以更常用于知识表示。



下面利用积木世界的例子说明如何使用一阶逻辑表示知识。假定桌上有三块积木,顶上的积木为绿色,底下的积木不是绿色,中间的积木颜色未知。我们利用 a、b、c 分别表示积木,谓词 O 和 G 分别表示“在……之上”和“……为绿色”,因此上述情况可表示

为如下事实集合{O(a,b),O(b,c),G(a),¬G(c)},于是我们能调用**推理机**判断出$\exists x\exists y G(x)\wedge\neg G(y)\wedge O(x,y)$不成立,即不存在一个非绿的积木直接在绿色积木上面。

面对复杂多变的知识处理问题,标准逻辑不断暴露出种种局限性。于是,自20世纪以来,针对传统形式逻辑的不足,先后提出了多种新的逻辑体系,统称非标准逻辑或现代逻辑。例如,针对传统形式逻辑是精确的、非此即彼(非真即假)的二值逻辑,发展了除真假值外,还可取未知或既真又假值等的多值逻辑和按隶属函数在0(假)和1(真)之间取任意值的**模糊逻辑**;针对传统形式逻辑与情况变化或时间无关的局限,发展了情境逻辑和时态逻辑;针对传统形式逻辑的单调性局限,即其公理系统(**知识库**)的一致性会因加入新的真命题(知识)而遭破坏,发展了**非单调逻辑**;针对传统形式逻辑不能表达事物或其属性的必然性、可能性(或然性)、应当、也许、允许、禁止、可信、知道等模态命题或模态演算,发展了模态逻辑。现有的非标准逻辑远不止这些,而且新的非标准逻辑还在不断产生,其目的都是为了扩充逻辑的表示能力或提高其推理效能。

逻辑表示是一种**陈述性表示**方法,具有陈述性表示的所有优点。自从知识表示研究领域出现以来,人工智能界对逻辑是否为最佳的知识表示方法一直争论不休。支持者(包括 McCarthy, Hayes, Moore 等人)认为一阶逻辑及其片段具有良好的表达能力、完善的语义,以及较强的推理能力,非常适于表示知识和捕捉推理。反对者(包括 Bar-Hillel, Minsky, Brooks 等人)主要基于如下四点进行了反驳:仅仅使用演绎推理有时很难完全满足实际需求;通常进行演绎推理的代价太高;记录下所有知识是不可行的;在一些领域,其他方法表现更好,例如**机器学习**领域的**统计学习**。无论争论结果如何,一个不容忽视的事实是,逻辑表示方法已应用于人工智能领域的几乎所有分支。

总之,在对理论进行深入研究的基础上,如何扬长避短、增强表达能力、提高推理效率是逻辑表示研究的永恒主题。从应用的视角来看,针对不同应用领域的知识特点,如何将逻辑表示与面向对象、框架、函数和过程等知识表示方法更好地结合起来,力求在增强表达能力的同时,提高推理效率,将是重要的发展方向。

参考文献

1. van Harmelen F, Lifschitz V, Morgenstern L, Plaisted D, Porter B, eds. Handbook of knowledge representation. Elsevier, 2008

2. Brachman R J, Levesque H J. Knowledge representation and reasoning. Morgan Kaufmann, 2004

(刘大有 欧阳继红 童频)

luoji chengxu sheji

逻辑程序设计(logic programming) 编写逻辑程序的方法与过程。逻辑程序由单元逻辑子句(也称事实,例如:p(x))和条件逻辑子句(也称规则,如:p(x)←q(x),其中p(x)为子句头,q(x)为子句体)组成。逻辑程序设计的主要任务在于用单元子句和条件子句描述领域知识中的事实和规则,作为求解问题的前提,然后根据推理规则求解问题。

逻辑程序设计源于法国数学家 J. Herbrand 的开创性工作。早在1930年,Herbrand 就证明了可以用机械的方法判定逻辑子句的可满足性。1965年,美国 J. A. Robinson 提出了面向计算机的合一算法和归结原理。1971年,R. Kowalski 在上述工作的基础上,论证了一阶谓词逻辑的子集(即 Horn 子句逻辑)可以解释递归过程,从而开辟了逻辑式程序设计的新领域。遵循 Kowalski 的路线,法国马赛大学的 A. Colmerauer 等人首先用 **FORTRAN 语言**实现了一个自动但效率较低的证明程序,并命名为 Prolog。70年代后期,D. H. D. Warren 等人在英国爱丁堡大学实现效率较高的编译型 Prolog 版本。80年代至今,逻辑程序设计沿着并行化和约束化两大方向继续发展。

逻辑程序设计的一个重要方面是 Horn 逻辑程序设计。Horn 逻辑程序由 Horn 子句(即不带否定的谓词公式最多不能超过一个,例如:p(x)←q(x),其中p(x)为子句头,q(x)为子句体)组成。Horn 逻辑程序设计的基本内容包括如何构造数据结构,定义程序和求解问题等。Horn 逻辑程序中数据结构统一为项。项是树形数据结构,可以表达任意复杂的数据对象。最简单的项是逻辑常量与逻辑变量。操作数据结构的唯一方法是通过合一操作,合一操作递归地通过项的等同比较,或逻辑变量例化,使得被操作的项变成等同项。它具有变量赋值,结构匹配,结构分解与合成等多种功能。通过定义表达事实的单元子句和表达规则的条件子句定义程序。通过提出问题启动和执行程序。程序的执行过程为 SLD 归结过程。SLD 是 Robinson 一般归结方法在

Horn 逻辑条件下的精化。它是一种反驳式的推理方法。推理序列是逐步递归地构成的,最初结点为原始目标或问题,它是推理序列的起点。以后每一步均由一种计算规则 R 规定从目标中选取一个子目标。使用此子目标和逻辑程序中所有可能的 Horn 子句头合一,实施一步 SL 归结,把产生的归结式作为推理序列的新结点。重复上述过程,直至产生空结点或归结失败为止。

Horn 子句和一般子句的差别在于一般子句允许有否定条件(例如,p← ~q,r)。在 Horn 子句逻辑中,不能直接证明一个否定目标。一种特殊的方法是 1978 年 K. L. Clark 提出的有限失败假设,即对于归结失败的目标(如(p(x)),基目标的否定(如 ~p(x))成立,它是对 SLD 的扩充,称为 SLDNF。逻辑程序设计的一个显著特点是具有非过程性:所有待解问题均是逻辑程序本身的逻辑推论,即任何满足逻辑程序的模型均满足该推论。这就表明,问题求解与程序书写的次序无关,与具体执行过程无关。按照 SLD 推理方法,逻辑程序的解空间有不同的描述方法,例如 OR 树方法,AND 树方法,AND/OR 树方法。它们在刻画和描述程序并行性方面虽有差别。但目的都是为了开发逻辑程序本身存在的各种并行性,主要有子目标的与并行,选择子句的或并行及含公共变量两个子目标之间的流并行,即一个子目标作为生产者,一个子目标作消费者之间的并行。

逻辑程序设计应用:由于逻辑程序设计使用更加面向人类的逻辑语言,以及逻辑程序蕴含的推理机制,使得逻辑程序设计方法获得广泛的应用。逻辑程序设计提供了有关启发式搜索,问题归结和问题求解等简单而有效的模型。逻辑程序设计与语言学的结合产生逻辑文法就是一个典型的例子。逻辑程序设计为专家系统提供单一的简单的知识描述语言,逻辑程序设计语言本身提供了知识库管理机制和使用知识的推理机制,以及专家系统中常见的双向匹配和回溯机制。

参考文献

1. Kowalski R. Logic for problem solving. Elsevier North Horlans Inc. ,1979

2. Amble T. Logic programming and knowledge engineering. Addison-Wesley Publishers Ltd, 1981

3. Bowen A K. Prolog and expert systems. McGraw-Hill Inc. ,1991

4. Walker A. Knowledge systems and Prolog. Addison-Wesley Publishing Company Inc. ,1987

(胡运发)

luoji chengxu sheji yuyan

逻辑程序设计语言(logic programming language) 用于**逻辑程序设计**的语言。组成逻辑程序的语句的基本形式是 Horn 子句,其形式为

A if B1 and B2 and. . . and Bn

其中 A 是原子公式作为结论,零个或多个原子公式的合取作为条件。若其中任一 Bi(1≤i≤n)要么是原子公式,要么是原子公式的否定,则称为规范形式。若其中 Bi(1≤i≤n)可以是任意一阶逻辑公式,则称为一般形式的子句。J. W. Floyd 和 R. W. Topor 已经指明任意一般形式的逻辑子句均可转化为规范形式子句。

逻辑程序设计语言有:①顺序逻辑程序设计语言;②并行逻辑程序设计语言;③约束逻辑程序设计语言。顺序逻辑程序设计语言的代表是 **PROLOG 语言**。Prolog 一个显著的特点是其执行过程有明显的顺序性:子目标 Bi(1≤i≤n)执行顺序是从左向右,选择适用子句的次序是从上向下,搜索策略是深度优先。在单中央处理器(CPU)计算机上,顺序逻辑程序设计语言有较高的执行效率,缺点是求解机制不够完备。

并行逻辑程序设计语言的典型代表有 K. L. Clark 提出的 **PARLOG 语言**。其特点是并行执行所有与目标 Bi(1≤i≤n),并且对于所有满足条件的子句要进行选择提交。推理过程中,一直向前,没有回溯,此语言在多 CPU 机和多机环境下具有较高执行效率和并发通信能力。

约束逻辑程序设计语言(CLP)是在顺序或并行逻辑程序设计语言中增加一些特殊原语和推理方法而形成的语言。在逻辑程序执行过程中,这些原语可以自动被延迟或被调用,以便多模式地求解问题。例如求解简单方程的原语,可求解方程 $X=Y+4$,如 X,Y 未例化,则该方程被延迟,一旦 X,Y 中有一个被例化,则 $X=Y+4$ 立即被求解。典型的约束逻辑程序设计语言有 PROLOG Ⅲ,Chip,CLP(R)等。对某些应用,比如涉及方程求解的搜索问题时,通过大量方程的求解,可删除不必要的搜索空间,从而大大提高问题的求解速度。

参考文献

Kowalski R A. Predicate logic as programming language. In: Proc. IFIP 74. Amsterdam: North Holland

Publishing Co.,1974:569-574 （胡运发）

luoji jicheng dianlu

逻辑集成电路（logic integrated circuit） 基于布尔代数（逻辑代数、开关代数）的公式及规则，能对二进制数进行布尔运算的集成电路。由于逻辑集成电路的输入和输出都是离散的物理量（如高电平为“逻辑1”、低电平为“逻辑0”），又称为**数字集成电路**（数字IC），以区别于对连续物理量进行操作的模拟集成电路。数字IC主要是指由门电路和记忆电路组成、且能实现一定逻辑功能的集成电路，这些逻辑功能包括数字逻辑运算、存储、传输及转换等。数字IC是数字电子计算机和所有数字电子系统的硬件基础。

数字IC按其内部有源器件的不同可分为两类：一类为双极型晶体管集成电路；另一类为绝缘栅场效应晶体管集成电路（或称MOS型电路）。在工艺上，数字IC已从当初的电阻-晶体管逻辑（RTL）、二极管-晶体管逻辑（DTL）发展到了**晶体管-晶体管逻辑**（TTL）和**射极耦合逻辑**（ECL），同时出现了PMOS、NMOS、CMOS和GaAsMOS等MOS型数字IC。表1为主要数字IC类型及其电性能比较。

表1 主要数字IC类型及电路性能比较

电路类型	电路结构	功耗	速度特性	集成特性
RTL	双极型	高	低	分立
DTL	双极型	高	低	分立、SSI
TTL	双极型	中	中	SSI、MSI
ECL	双极型	高	高	SSI、MSI、LSI
PMOS	MOSFET	中	低	MSI、LSI
NMOS	MOSFET	中	中	MSI、LSI、VLSI
CMOS	MOSFET	低	中	SSI、MSI、LSI、VLSI、更高
GaAs	MESFET	高	高	SSI、MSI、LSI

数字IC主要由门电路组成，因之可按每个芯片上集成的等效门电路个数或元器件个数来表征该芯片的集成度。通常按集成度将数字IC分为6类，即**小规模集成电路**（SSI）、**中规模集成电路**（MSI）、**大规模集成电路**（LSI）、**超大规模集成电路**（VLSI）、**特大规模集成电路**（ULSI）和**巨大规模集成电路**（GLSI），如表2所示。作为比较，表中列出了晶体管和分立元件的发展年代。

表2 集成电路的集成度分类

年份	1948	1961	1966	1971	1980	1990	1998	2000
类别	晶体管	分立元件	SSI	MSI	LSI	VLSI	ULSI	GLSI
芯片所含等效门电路个数	1	1	小于10	$10\sim10^2$	$10^2\sim10^4$	$10^4\sim10^6$	$10^6\sim10^8$	大于10^8
芯片所含元器件个数			小于10^2	$10^2\sim10^3$	$10^3\sim10^5$	$10^5\sim10^7$	$10^7\sim10^9$	大于10^9
代表产品	二极管 三极管		门电路 触发器	计数器 加法器	8位微处理器	16位、32位微处理器	图像处理器，SOC文档微处理器	

按照芯片的设计方法，数字IC可分为通用集成电路（如市售的小、中、大规模集成电路产品）、可编程逻辑器件（如PROM、EPROM、FPLA、PAL等）、半定制集成电路（如门阵列、标准单元等构成的集成电路）和全定制集成电路。

按其输出的生成方法，数字IC可分为组合（逻辑）电路和时序（逻辑）电路两种。组合电路的输出只取决于当前输入的组合，不受过去输入的影响；时序电路的输出则取决于当前输入和过去输入所引起的当前状态。

数字IC分别以高、低电平表示逻辑1和0。若IC用高电平表示逻辑1，低电平表示逻辑0，称为正逻辑电路；反之，高电平对应逻辑0，低电平对应逻辑1的电路称为负逻辑电路。这是在设计数字IC时约定的。

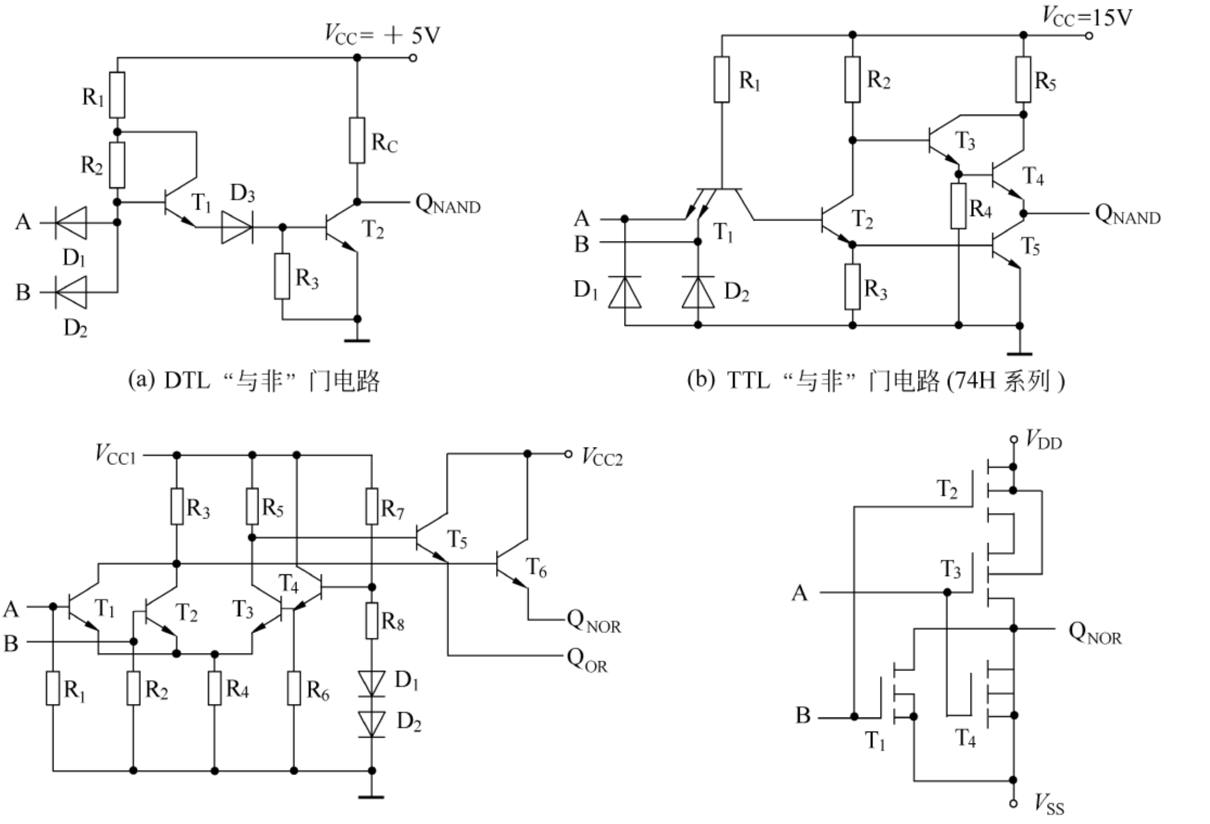
门电路是组合电路中的基本电路，它按照输入端

条件产生输出。基本门电路有“与”门、“或”门、“非”门、“与非”门、“或非”门、“与或非”门、“异或”门、“同或”门等。表 3 中以 A，B 双输入和 Q 输出为例给出基本门电路的逻辑表达式、电路符号表示和真值表。

表 3　基本门电路的逻辑表达式、电路符号表示和真值表

逻辑门名称	“与”门 AND	“或”门 OR	“非”门 NOT	“与非”门 NAND	“或非”门 NOR	“异或”门 EOR	“同或”门 ENOR
逻辑表达式	$Q=AB$	$Q=A+B$	$Q=\bar{A}$	$Q=\overline{AB}$	$Q=\overline{A+B}$	$Q=\bar{A}B+A\bar{B}$	$Q=AB+\bar{A}\bar{B}$
电路符号表示	A, B — & — Q	A, B — ≥1 — Q	A — 1 —○ Q	A, B — & —○ Q	A, B — ≥1 —○ Q	A, B — =1 — Q	A, B — = — Q
真值表	A B Q 0 0 0 0 1 0 1 0 0 1 1 1	A B Q 0 0 0 0 1 1 1 0 1 1 1 1	A Q 0 1 1 0	A B Q 0 0 1 0 1 1 1 0 1 1 1 0	A B Q 0 0 1 0 1 0 1 0 0 1 1 0	A B Q 0 0 0 0 1 1 1 0 1 1 1 0	A B Q 0 0 1 0 1 1 1 0 0 1 1 1

图 1 给出了四种典型电路图。以图 1(b)为例，当 A，B 同时为高电平时，T1 截止，T2 导通，导致 T3、T4 截止，T5 导通，输出为低电平；当 A，B 中有一个为低电平时，T1 导通，T2 截止，导致 T3、T4 导通，



(a) DTL “与非” 门电路　(b) TTL “与非” 门电路 (74H 系列)

(c) ECL “或非”“或” 电路　(d) CMOS “或非” 门电路

图 1　基本逻辑门电路图

T5 截止,输出为高电平。

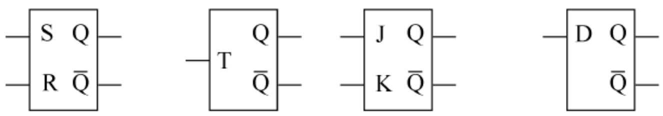
具有复杂功能的组合电路可由上述的基本门电路构成。由于中、大规模集成电路技术的进步和成本的降低,可以把由许多个基本门电路构成的组合电路作为基本 IC 使用。这类组合电路有译码器、编码器、数据选择器、数值比较器、模拟开关、**算术逻辑部件**、奇偶产生器、奇偶检验器、**只读存储器**和可编程组合逻辑器件(如 PLA、PAL、FPLA)等。

触发器是时序电路中构成存储电路最常用的存储元件,它的输出不仅依赖于当前输入端的状态,而且依赖于当前该电路的内部状态,而该内部状态又依赖于前一时刻电路输入端的状态和前一时刻的内部状态。基本的触发器电路有 R-S 触发器、T 触发器、J-K 触发器和 D 触发器。图 2 是 R-S 触发器电路图,它由两个"与非"门或者两个"或非"门电路加反馈构成。门的输入端分别为复位 R、置位 S,输出端为 Q 和 $\overline{Q}$。当 S = 1,R = 0 时,Q = 1(置位状态);当 S = 0,R = 1 时,$\overline{Q}$ = 1(复位状态);R = S = 0 时,保持原状态不变;R = S = 1 时,Q,$\overline{Q}$ 状态不定。图 3 为上述 4 种触发器的电路符号和工作特性。



(a) 用"或非"门组成的高电平触发的 R-S 触发器　(b) 用"与非"门组成的低电平触发的 R-S 触发器

图 2　基本 R-S 触发器电路图



R	S	Q_{n+1}
0	0	Q_n
0	1	1
1	0	0
1	1	不定

(a) R-S 触发器

T	Q_{n+1}
0	Q_n
1	$\overline{Q}_n$

(b) T 触发器

J	K	Q_{n+1}
0	0	Q_n
0	1	1
1	0	0
1	1	$\overline{Q}_n$

(c) J-K 触发器

D	Q_{n+1}
0	0
1	1

(d) D 触发器

图 3　基本触发器的电路符号及其工作特性

具有复杂功能的集成时序电路主要有寄存器、移位寄存器、计数器、可读写存储器(RAM)和可编程时序逻辑器件(如 RPLA、GAL、FPGA)等。寄存器和移位寄存器都是计算机中用来暂时寄存数码的存储元件,它们都具有接收、存储和传送数码的功能。移位寄存器还具有移位功能,可以单向移位或双向移位。按寄存器和移位寄存器的工作方式和输出稳定状态可分为静态和动态两种;若按输入、输出形式可分为串入-并出、串入-串出、并入-串出和并入-并出四种。计数器是计算机中广泛应用的一种可以记录脉冲个数的电路。计数器按记数制可分为二进制计数器、十进制计数器、二-十进制计数器等;按其计数功能可分为加计数器、减计数器和可逆计数器;按预置和清除方式则可有并行预置、直接预置、同步清除和异步清除。计数器还可按工作方式分类,有同步计数器和异步计数器。

由于数字 IC 容差大、体积小、重量轻,因而得到了迅速发展,其应用越来越广泛,内容越来越丰富,技术越来越成熟。在许多情况下,完成同样的功能,数字 IC 与模拟 IC 相比,速度更快、精度更高、工作更稳定、抗干扰能力强、实现更容易、操作更简便,此外在信息的处理、储存、故障检测与校正等方面更加方便。数字 IC 正朝着更加高速、低耗、低电源电压和高集成度方向发展。

参考文献

1. 尤忠琪,贾立新. 数字集成电路教程. 北京:科学出版社,2001

2. 杨文霞,孙青林. 数字逻辑电路. 北京:科学出版社,2007

(时万春)

luoji tuiliji

逻辑推理机(logic inference machine) 可自动进行推理的计算机。它的输入是所要求证明的推理目标、有关的变量以及前提和假设;输出是关于推理目标的证明结论、有关的解释以及上述变量的值。程序员为逻辑推理机编写逻辑程序,其中包括推理过程中需要的已知事实,表示推理规则的逻辑语句和其他形式的语句等。通常,逻辑程序是用 PROLOG 程序设计语言编写的。逻辑推理机根据上述信息,按照其本身所具有的推理机制,选择适当的决策步骤进行逻辑推理演算并输出结果。

逻辑推理机必须是完善的,即不可能给出错误的推理结果,又必须是完备的,即能够给出全部正确推理结果。

逻辑推理机的体系结构可以用 Warren 抽象机(WAM)语义模型表示。WAM 模型的指令集包含以下 5 类基本指令:

(1)索引类指令 在一个子句的推理过程中,用于控制各部分执行次序,例如,选择当前子句、回溯、重试等;

(2)过程类指令 用于管理推理过程中的子句链的选择和环境设置,以及子句链之间的转移等;

(3)取检类指令 包括取出参数,检验子句的形式参数与实在参数是否合一,记录相应的变量代入关系等;

(4)设置类指令 在子句体中,为谓词设置实在参数等;

(5)合一类指令 用于处理表或函数结构中各分量的设置和合一操作等。

应用这些指令,可以编写相应的程序来描述逻辑推理机的体系结构;又可以通过它们,采用编译、解释以及硬件组成等方法去实现实际的逻辑推理机。

关于逻辑推理机的研究成果有美国的 PLM 机(1985 年)、日本第五代计算机计划中的 PSI 机(1985 年)和 PSI-Ⅱ机(1987 年)以及具有并行化体系结构的 Multi-PSI 机(1987 年)和 PIM 机(1986 年)等。

(郑守淇)

luojixue

逻辑学(logic) 研究思维推理形式与方法的学科,亦常简称为“逻辑”。旧译有“名学”、“辩学”、“名辩学”、“名理学”、“理则学”、“论理学”等。通常有狭义与广义的不同理解。就基础理论而言,狭义的逻辑学仅指演绎逻辑学,广义逻辑观还包括归纳逻辑学与辩证逻辑学。

国际学界公认逻辑学思想有三大源头:古希腊的推理与论证学说、古印度的正理-因明学说和中国先秦的名辩学说;但只有古希腊的推理与论证学说从论辩学与认识论中独立出来,形成了比较纯粹的逻辑理论。古希腊哲学家亚里士多德被公认为逻辑学之父,其著作《工具论》确定了后世逻辑学研究的主要畛域。

17 世纪德国学者哥特弗雷德·威廉·莱布尼茨,最早提出了系统运用数学方法研究逻辑问题的设想,并完整地提出了使用“通用语言”进行“推理演算”的研究纲领。莱布尼茨研究纲领提出近二百年之后,19 世纪英国学者乔治·布尔提出的“逻辑代数”。布尔发现,命题之间的逻辑关系与某些数学运算很相似,代数系统可以有不同的解释,将之推广到逻辑学领域,就可以构成一种思维演算。1879 年德国学者戈德罗布·弗雷格建构了构成现代演绎逻辑基础的命题逻辑与谓词逻辑系统,弗雷格被公认为现代演绎逻辑最重要的奠基人。英国学者阿尔弗雷德·怀特海与罗素合著的《数学原理》(1910—1913 年),在完善与传播现代演绎逻辑基础方面发挥了至关重要的作用。

第一个运用形式系统方法研究模态逻辑、构造现代模态逻辑系统的是美国学者克拉伦斯·欧文·刘易斯,其方法是在经典逻辑的基础上引入“必然”“可能”这两个模态算子和关于它们的公理与规则,来建构各种模态逻辑形式系统;到 20 世纪中期,美国学者索尔·克里普克等人创建了“可能世界语义学”,使现代模态逻辑得以确立。这些成果继续鼓舞了逻辑学家们把研究向“广义模态逻辑学”扩张,即在经典逻辑学基础上,通过引进时态算子(“过去”“现在”“将来”等)建立“时态逻辑学”,引进认识论算子(“知道”“相信”等)建立“认识论逻辑学”,引进道义算子(“应当”“允许”等)建立“道义逻辑学”,如此等等,形成了一个庞大的新型学科群。由于这些新算子都来自哲学中的一些基本概念或范畴,所以被广泛地称为“哲理逻辑学”或“哲学逻辑学”。这种意义上的“哲理逻辑学”各分支有一

个共同的特点,就是它们皆为经典逻辑基础上的“保守扩张”,即都是在承认经典逻辑学的基础上,通过引入新的哲理性算子构造逻辑系统,探究基于这些算子的逻辑推理机理。但是,许多哲学家和逻辑学家指出,相对于人类实际思维而言,经典逻辑学本身具有“高度理想化”的特点,虽然这是科学抽象难以避免的,但逻辑学研究也应当反过来逐步逼近人的实际思维,沿此思路又产生了各种“异常逻辑学”。各种异常逻辑学理论的构建背景,都在某些关键点上“异于”经典逻辑学的基本理念,比如异于经典逻辑学的二值性而建构“多值逻辑学”,异于经典逻辑学之谓词精确性而建构“模糊逻辑学”,异于经典逻辑学之实质蕴涵理论而建构“相干逻辑学”,异于经典逻辑学“个体域非空”和“专名非空”假设而建构没有这种假设的“自由逻辑学”,甚至建构不承认“排中律”的“直觉主义逻辑学”和不承认“矛盾律”的“亚相容逻辑学”(又译“次协调逻辑学”、“弗协调逻辑学”),如此等等。特别地,在当代计算机科学与人工智能研究的推动下,异于经典逻辑学之“单调性”的各种“非单调”动态逻辑学研究亦蓬勃兴起。而这些变异逻辑系统也被实施扩充,从而形成“多值模态逻辑学”、“亚相容模态逻辑学”、“非单调模态逻辑学”等“变异扩充”理论。由于这些“变异”都基于一定的哲学考虑,许多学者也把“变异逻辑学”学科群称为另一大类“哲理逻辑学”。这两大类哲理逻辑学研究在20世纪后半期形成了研究热潮,出现了许多学派,但由于它们具有共同的形式系统方法,又具有共同的演绎有效性诉求,因而可以展开富有成效的研究对话,极大地推进了对人类实际演绎推理机理的认识与把握,使得演绎逻辑学研究获得空前繁荣,构成了作为基础学科的当代逻辑科学的主体部分。

现代演绎逻辑方法的实质不在于使用“数学方法”,而在于在严格区分“思维形式”与“思维内容”的亚里士多德传统之上,进一步建构能够严格区分思维形式之“语法”与“语义”的形式系统,从而可以严格地研究系统的语形学、语义学及其相互关系,这使得彻底严格的“元理论”研究成为可能。这种研究不仅可以实施于逻辑系统本身,而且可以实施于任何可以公理化的非逻辑理论,只要我们把理论的公理形式化,同时又使用形式化的逻辑工具,那么就可以构建该理论的形式系统,继而研究系统的相容性、完全性等“元理论”性质。分别侧重于研究形式系统的语形学与语义学的“证明论”与“模型论”,是现代演绎科学方法论的两大主要分支。它们与研究能行可计算理论的“递归论”,以及在解决素朴集合论悖论过程中形成的、作为现代形式化公理系统典范的“公理集合论”,通常被并称为狭义“数理逻辑学”(广义数理逻辑学包括全部现代演绎逻辑学)。该“四论”研究产生了哥德尔不完全性定理、塔尔斯基形式语言真理理论、邱奇不可判定性定理等划时代理论成果,并通过递归论的中介,在当代计算机科学与人工智能研究中发挥着基础作用。在现代学科分类体系中,“四论”经常被归到“数学基础”研究之下,但它们又都具有一般哲学与方法论价值,属于当代逻辑学与数学学科的交叉研究领域。

归纳逻辑学是以归纳推理的可靠性为主要研究对象的学说。归纳推理是一种“或然性”推理,即在前提与结论间无法建立演绎保真关系的情况下,使前提为结论提供一定程度的支持。归纳逻辑学旨在为提高归纳推理前提对结论的支持度或推理的可靠性程度提供逻辑工具。归纳逻辑学思想亦可追溯到古希腊时期。在《后分析篇》中,为回答一个知识系统之中不能被演绎证明的“基本前提”或“公理”之合理性由何保证的问题,亚里士多德提出了“简单枚举归纳”和“直觉归纳”的基本思想。中世纪后期逻辑学研究中也拥有丰富的归纳逻辑思想。系统严整的归纳逻辑理论的创立者,是文艺复兴后期的英国学者弗兰西斯·培根,他在《新工具》一书中完整地阐述了作为观察与实验方法的“排除归纳法”。19世纪英国学者约翰·斯图亚特·密尔发展与完善了“排除归纳法”,在《逻辑体系》一书中确立了“探求因果联系的五种归纳方法”,同时对“类比推理”这种或然推理形式及其作用也予以了系统把握,从而成为传统归纳逻辑学的集大成者。传统归纳逻辑学与传统演绎逻辑学一起,在近代实验科学的创立与发展中起了极其重要的作用。

现代归纳逻辑学的发展,也是20世纪逻辑学发展高峰的一个重要侧面。其特点是依托现代演绎逻辑学的长足发展,在与演绎逻辑学的互动中展开研究。20世纪前半期归纳逻辑学研究主流的特点,是将其研究重心从传统归纳逻辑学关于“科学发现”(假说之提出)的归纳机理研究转移到“科学检验”(假说之验证)的归纳机理研究,其显著标志是概率工具的引入和系统运用。实际上,在培根的《新工具》出版约40年之后,法国数学家布雷斯·帕斯卡等就已通过赌博中的“盖然性”的量化研究制订了概率演算的基本原则,此后莱布尼茨等人也对此做

了理论与应用研究(包括在法庭论证与决策中的应用),布尔也曾试图把他的逻辑代数做概率解释,但他们都没有将概率演算引入归纳逻辑学研究。直到20 世纪 20 年代初,才由英国学者约翰 · 凯恩斯对概率概念做了“逻辑解释”,并将之系统地引入归纳逻辑学研究。此后,德国学者鲁道夫 · 卡尔纳普等人运用现代演绎逻辑的形式系统方法,建构了关于概率归纳演算的形式系统,以应用于科学验证(“证据对假说的归纳支持”)之“确证度”的量化研究。20 世纪后半期迄今,“发现的逻辑”研究在新的基础上得到恢复与发展,特别体现在运用现代哲理逻辑的成果提出探求因果联系的新理论,而概率归纳逻辑学研究出现了所谓“非帕斯卡方向”的“新培根主义”理论,表现为对统一刻画“发现”与“验证”中的逻辑机理的诉求。归纳与演绎在人类实际思维中的互补机理,在这种新的探索中得到了更好地揭示。

辩证逻辑学是一种以哲理范畴推演为核心的逻辑学说。其系统研究发端于亚里士多德的《形而上学》及《工具论》中的《范畴篇》、《论辩篇》,作为对象明确的“逻辑类型”成型于康德“先验逻辑学”及黑格尔“思辨逻辑学”的“辩证转换”,经过马克思主义者“祛魅”而明确其科学方向。18 世纪德国学者伊曼努尔 · 康德把演绎逻辑学与归纳逻辑学统称为“形式逻辑学”(这个称呼得到了广泛采纳),他发现,在形式逻辑学所“普适”但不研究的“思想内容”方面,实际上存在着为人们长期忽视的一种重要的层面区分:经验内容和先验内容。思想的经验内容是可以通过观察与实验方法把握的,但制约这种把握的不仅有演绎与归纳的“形式”,还有一种既不是思想的“形式”也不是思想的“经验内容”的东西,比如亚里士多德的《范畴篇》中“实体”“性质”“关系”等范畴及其相互作用的内容,它们既不属于形式逻辑的“形式”,也不属于可以经验验证的“经验内容”,而属于“先验内容”或“纯内容”。这种“纯内容”,表现在思维中就是作为“纯概念”的哲理范畴。正是制约它们的法则(连同形式逻辑法则一起)构成了科学知识之“必然性与普遍性”何以可能的条件。这就是康德所谓“先验逻辑学”的研究对象。19 世纪德国古典哲学的集大成者乔治 · 威廉 · 弗里德里希 · 黑格尔肯定了康德关于先验范畴及其对求真讲理之特殊重要性的认识,又指出康德之所以因其先验逻辑研究导致“二律背反”而陷入不可知论,是因为他只是静态的、固定的把握“先验范畴”,而如果以动态的、流动的观点来把握这些范畴,不但“二律背反”是可解的,而且可以产生一种具有重要的方法论意义的新的逻辑类型,即“思辨逻辑学”或“辩证逻辑学”。黑格尔把固定范畴转化为流动范畴的关键环节,是通过对康德“二律背反”理论的改造,提出了“辩证否定”和“辩证矛盾”学说,并建构了以此为轴心的动态化范畴体系。这个范畴体系的建立,是人类对辩证思维方法的把握从自发的素朴形态上升为自觉的理论系统形态的一个标志。黑格尔的辩证逻辑,在马克思主义创立与发展的过程中起到了特殊的作用。马克思、恩格斯及列宁都提出了建构祛除黑格尔神秘色彩的科学形态的辩证逻辑学的设想。

随着现代逻辑学基础理论的发展,逻辑应用研究也获得了空前广泛的展开。现代逻辑学的应用不仅改变了哲学研究的面貌,导致了哲学研究的“语言论转向”,也改变了许多学科乃至现代科学技术整体发展的风貌。20 世纪前半期语言学中乔姆斯基生成转换语法,心理学中皮亚杰的认识发生学,特别是导致当代信息革命的计算机技术的诞生等,都是运用现代逻辑最新成果的产物。以系统论、信息论、控制论为先导的当代系统科学的出现,也与现代逻辑学发展中提供的新工具密切相关。20 世纪后期以来,现代逻辑学的应用更是形成了遍地开花的局面,其理论与方法不同程度地渗透到几乎所有学科领域之中。与此同时,也形成了作为逻辑基础理论与逻辑应用研究之“中介”的一系列逻辑应用理论,可称之为“应用逻辑学”学科群。

“科学逻辑学”(logic of science)是逻辑应用理论的一个范例。现代归纳逻辑的代表人物卡尔纳普首先使用了“科学逻辑学”这一学科称谓,用以指谓演绎逻辑与归纳逻辑在科学理论结构中的作用机理研究。这种用法被后人发展为对如下研究领域的称谓,即逻辑因素在科学研究各环节(科学发现、科学说明、科学验证及科学演进)的作用机理,以及逻辑因素与非逻辑因素相互作用机理的系统探究与把握,即在科学研究中的逻辑应用方法论模式研究。科学逻辑学研究的始祖,可追溯到亚里士多德《工具论》之《后分析篇》。《后分析篇》可视为第一个系统的科学逻辑文本。尽管其主体是演绎科学方法论,但也建立了历史上第一个以归纳 - 演绎程序为中介、以观察和解释性原理为两翼的逻辑应用方法论体系。当代科学逻辑可以视为亚里士多德全面探讨科学研究中的逻辑应用方法论之诉求的当代后裔。“非形式逻辑学”(informal logic)是 20 世纪后

期兴起研究热潮的另一逻辑应用理论。其研究诉求，就是要系统把握逻辑因素在日常非形式论证与批判性思考中逻辑应用方法论模式，亦即系统把握逻辑因素在非形式论证与批判性思考中的作用机理，以及逻辑因素与非逻辑因素在其中的相互作用机理，因而又被称为“论辩逻辑”(logic of argumentation)。亚里士多德《工具论》中的《论辩篇》，中国先秦经典《墨辩》，古印度经典《正理经》，都可视为这种逻辑应用理论的先驱。与该分支密切相关，亦在20世纪后期兴起研究热潮的自然语言逻辑(logic of natural language)，则是发展比较成熟的另一逻辑应用理论，其宗旨主要在于刻画逻辑因素在语言交际过程中的逻辑应用方法论模式，这一分支的发展推动了当代逻辑学与语言学研究的“语用学转向”。逻辑学基础理论在作为当代“大科学”的认知科学中的广泛应用，也催生了作为认知领域的逻辑应用理论的“认知逻辑”(logic of cognition)或“心智逻辑”(logic of mind)的出现。认知逻辑是连接基础逻辑与当代人工智能研究中的逻辑应用的重要桥梁。除上述已获得较大发展的分支外，逻辑应用理论的一些新的分支也正在兴起，如“博弈-决策逻辑”“法律逻辑”“教育逻辑”等。

在逻辑学发展史上，除逻辑基础理论与应用理论之外，一些逻辑学的“外围学科”(可统称为“逻辑学学”)发挥着特殊功能，如逻辑史学、逻辑哲学、逻辑社会学、逻辑文化学等。逻辑哲学的真理理论、意义理论和逻辑悖论研究，在逻辑学发展史上都扮演了重要角色。近年兴起的逻辑社会学与逻辑文化学研究，致力于探讨逻辑学的社会文化功能及其作用途径，进一步推动了逻辑理论与应用研究在广度与深度上的发展。

参考文献

1. 周礼全. 逻辑百科辞典. 成都：四川教育出版社，1994

2. 彭漪涟，马钦荣. 逻辑学大辞典(修订本). 上海：上海辞书出版社，2010

3. 王习胜，张建军. 逻辑的社会功能. 北京：北京大学出版社，2010

(张建军)

luoji yunsuan

逻辑运算(logic operation) 对有且仅有两种逻辑属性(“是”和“非”或“真”和“假”)的变量以及由这种变量组成的逻辑函数所进行的运算。相应的学科是逻辑代数，是英国的布尔(G Boole)于1849年提出的，所以也叫布尔代数。

计算机常用的3种基本逻辑运算为：“与”(逻辑乘，符号为 · 或∧)、“或”(逻辑加，符号为 + 或∨)和“非”(求反，符号为 $\overline{\ }$，即在变量的上面加一横杠)。

“与”操作：当变量 X 和 Y 均为“1”时，$X \cdot Y$ 才为“1”，否则为“0”。

“或”操作：当变量 X 和 Y 任一(或同时)为“1”时，$X+Y$ 为“1”，否则为“0”。

“非”操作：当变量 X 为“1”时，$\overline{X}$ 为0；当 X 为“0”时，$\overline{X}$ 为“1”。

以上3种操作的变量可扩大到两个以上，同时还可以用表达式来替代变量。

从这3种基本逻辑操作，可构造出任意逻辑函数。

在计算机中，往往设置有专门对两个操作数进行逻辑运算的指令，例如逻辑加、逻辑乘和异或(按位加)等指令，有时也可能要求其中1个数取反后再参与运算。逻辑运算是按位进行的，即每一位不受操作数中其他位的影响。表1是对两个数进行逻辑运算的**真值表**。

表1 逻辑运算真值表

A_i	B_i	C_i			
		逻辑加	逻辑乘	异或	同或
0	0	0	0	0	1
0	1	1	0	1	0
1	0	1	0	1	0
1	1	1	1	0	1

表中 A_i，B_i 分别为 A，B 两个操作数中的第 i 位，C_i 为运算结果 C 的第 i 位。

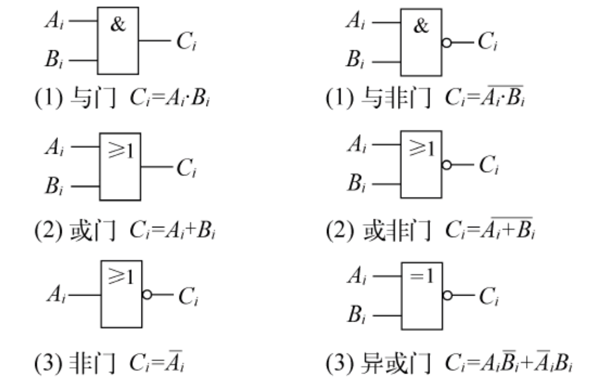
图1(a)是完成3种最基本逻辑运算的逻辑框图(即表示符号)以及相应的逻辑表达式。图1(b)是由基本逻辑电路组成的与非门、或非门、异或门的表示符号以及相应的逻辑表达式。

在计算机中央处理器的算术逻辑部件ALU中有完成逻辑指令功能的逻辑电路。除此以外，在计算机的各个部件中，还广泛将逻辑电路组合起来完成各种控制功能。

参考文献

王尔乾，杨士强，巴林凤. 数字逻辑及数字集成电路. 2版. 北京：清华大学出版社，2002

(王爱英)



(a) 基本逻辑电路框图　(b) 基本组合逻辑电路框图

图 1　逻辑电路框图

luoji zonghe

逻辑综合(logic synthesis)　将基于**硬件描述语言**的寄存器传输级电路描述转换为电路结构级描述(如门级网表)的过程。逻辑综合本质上是实现数字系统的逻辑设计及其优化的过程。数字系统逻辑设计的给定条件包括逻辑功能及逻辑实现的约束。逻辑功能由硬件描述语言(HDL)或布尔方程等各种形式给出。而逻辑实现的约束指逻辑单元的类型及其工艺条件等,如标准单元库。逻辑综合的目标是获得一个优化合理(成本最低或接近最低)的方案,即将一个用 HDL 给出的并经核实有效的设计描述,转换成由逻辑门构成的优化合理的网表描述,并同时产生一个用于验证和检测逻辑功能的完全测试集。

可以把逻辑综合的过程与编译高级语言程序的过程作一个类比。这里,基于硬件描述语言的 RTL 描述对应于用高级程序设计语言编写的源程序,标准单元库对应于目标机器的汇编指令集。逻辑综合的过程包括翻译(把 RTL 描述变换为中间表示)、优化(逻辑优化和最小化)、映射(选择合适的标准单元)这三个主要阶段,跟编译过程也是基本对应的。因此,也有人把逻辑综合称为硅编译。另外,从这个类比中也能体会出,逻辑综合的优劣对电路性能的影响以及逻辑综合的成熟对提高设计生产率的作用。

逻辑综合技术是**电子设计自动化**技术的一个重要组成部分。在逻辑优化的早期研究领域中,采用手工设计的方法来减少电路中门的数目,分别有**布尔代数**法、卡诺图法和真值表化简法(也称 Quine-McCluskey 法)。这些方法的致命弱点是无法综合规模较大的逻辑电路,输入变量不能多于 6～7 个。逻辑综合包括组合逻辑电路的综合和时序逻辑电路的综合。由于数字系统规模越来越大,功能更加复杂,而要求设计周期短,这就驱使人们采用计算机辅助设计的方法,即自动逻辑综合技术,完成整个系统的逻辑综合。超大规模集成电路的逻辑综合的目标有:芯片总布图面积最小化、关键路径时延最小化、可测试性最大化并辅助提供一个完备测试集。

逻辑综合主要包括如下内容:①逻辑描述及其编译　即将数字系统的逻辑功能描述转换成逻辑级的中间描述形式或逻辑电路的描述形式。②逻辑划分　即将所要设计的逻辑电路如何有效合理地分解成多个逻辑子电路,便于整体上更好处理并获得更优的结果。③状态化简　即在不改变系统性能前提下,将逻辑电路状态表的内部状态数减少到最少或接近最少的过程。④状态分配又称状态编码　就是将状态表中的各状态名分别赋予一个二进制值,使得状态表(即功能描述)可以用布尔表达式或方程组来表示。状态分配的好坏,直接决定了逻辑电路组合部分的复杂度。⑤逻辑优化分两级和多级逻辑优化　两级逻辑优化主要是指可编程逻辑阵列的逻辑最小化。它是在不改变逻辑功能的条件下,采用因子提取、无关项合并等方法,寻求一个具有最少变量和最少乘积项数的逻辑表达,从而降低组合逻辑的复杂度。多级逻辑优化则侧重于采用适当增加逻辑电路级数的办法来降低复杂度,简化原电路,而且不改变其逻辑功能和限制条件,降低造价。它通常与状态化简和状态分配结合在一起考虑。

在普通的逻辑综合中,只考虑有限的工艺相关信息,如标准单元库中提供的单元静态延迟。但在深亚微米及纳米工艺条件下,连线长度等布局布线后才能得知的信息对单元的延迟和负载大小有很大的影响。近年来发展起来的物理综合技术不仅考虑单元的静态特性,而且综合考虑了布局布线后的工艺信息,从而使逻辑优化有可能得到更优的结果。

片上系统及含数十亿晶体管的大型设计的出现,产生了将电子系统级(ESL)系统行为特性描述自动转换为寄存器传输级(RTL)描述的强烈需求。高层次综合就是为了满足这个需求而出现的一个新的研究和应用领域。高层次综合也称为系统级综合或行为级综合,主要包括 ESL 语言翻译、数据流综合、控制流综合等三个阶段。翻译是指自动将数字系统的功能行为描述转换成有利于后续优化的中间表示格式,其中,表示数据与操作相关性的中间表示

称为数据流图，表示对操作的控制间相关性的中间表示称为控制流图。数据流综合又分为操作调度和资源分配两个部分，其中操作调度是把实现系统算法所需要的所有操作或运算，以较优的形式分配到各个控制步骤；资源分配是为每个操作指派对应的硬件资源，包括运算、存储、连线等，这里涉及通过资源重用以降低成本和通过资源复制以提高性能的利弊权衡。控制流综合则根据控制流图以及数据流综合对控制部分的要求，综合产生控制流的逻辑图，最后给出以 HDL 描述的设计输出。

高层次综合中涉及许多 NP 难的问题，这给综合过程所需的时间、综合结果的可用性带来了不利的影响。目前，虽然已经有一些高层次综合工具问世，但在实际芯片设计中用得还不多。可以预期，高层次综合技术的日益成熟，必将大大提高电子系统设计的效率。

参考文献

1. 刘明业. 数字系统设计自动化. 北京：电子工业出版社，1991

2. Himanshu Bhatnagar. 高级 ASIC 芯片综合. 2 版. 张文俊，译. 北京：清华大学出版社，2007

（董云耀　唐志敏）

luoxuan moxing

螺旋模型（spiral model）　瀑布模型与**演化模型**相结合，并加入两者所忽略的风险分析所建立的一种软件开发模型。该模型于 1988 年由美国 TRW 公司 B. 鲍姆（B. W. Boehm）提出。软件项目风险的大小作为指引软件过程的一个重要因素，引入这一概念有可能使得软件开发被看作一种元模型，因为它能包容任何一个开发过程模型。

*软件风险*是任何软件开发项目中普遍存在的问题，不同项目在此的差别是风险有大有小而已。在制订软件开发计划时，系统分析人员必须回答：项目的需求是什么，需要投入多少资源以及如何安排开发进度等一系列问题。然而若要他们当即给出准确无误的回答是不容易的，甚至几乎是不可能的。但系统分析员又不可能完全回避这一问题。凭借经验的估计给出初步的设想便难免带来一定风险。实践表明，项目规模越大，问题越复杂，资源、成本、进度等因素的不确定性就越大，承担项目所冒的风险也越大。总之，风险是软件开发不可忽视的潜在不利因素，它可能在不同程度上损害到软件开发过程和软件产品的质量。软件风险驾驭的目标是在造成危害之前，及时对风险进行识别、分析，采取对策，进而消除或减少风险的损害。

螺旋模型沿着螺线旋转，在笛卡儿坐标的四个象限上分别表达了四个方面的活动（参看图 1），即：

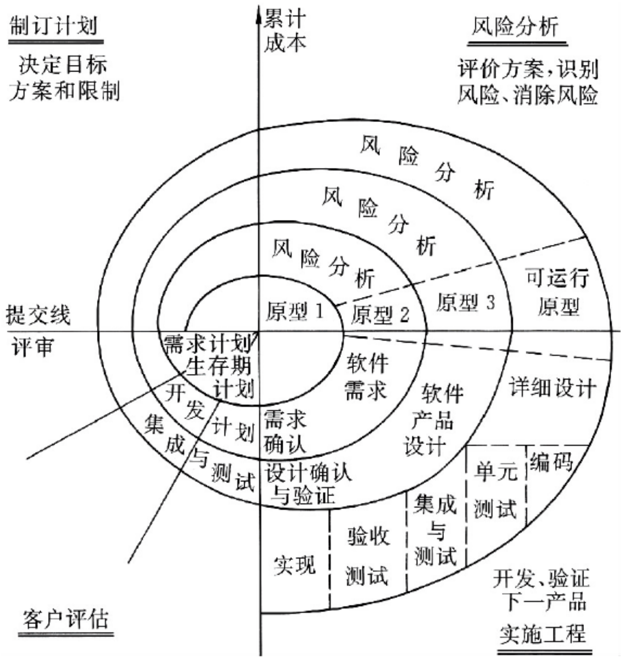


图 1　螺旋模型

（1）制订计划——确定软件目标，选定实施方案，弄清项目开发的限制条件；

（2）风险分析——分析所选方案，考虑如何识别和消除风险；

（3）实施工程——实施软件开发；

（4）客户评估——评价开发工作，提出修正建议。

沿螺线自内向外每旋转一圈便开发出更为完善的一个新的软件版本。例如，在第一圈，确定了初步的目标、方案和限制条件以后，转入右上象限，对风险进行识别和分析。如果风险分析表明，需求具有不确定性，那么在右下的工程象限内，所建的原型会帮助开发人员和客户，考虑其他开发模型，并把需求作进一步修正。

客户对工程成果作出评价后，给出修正建议。在此基础上需再次计划，并进行风险分析。在每一圈螺线上，风险分析的终点作出是否继续下去的判断。假如风险过大，开发者和用户无法承受，项目有可能终止。多数情况下沿螺线的活动会继续下去，自内向外逐步延伸，最终得到所期望的系统。图 2 给出了螺旋模型的另一图示。

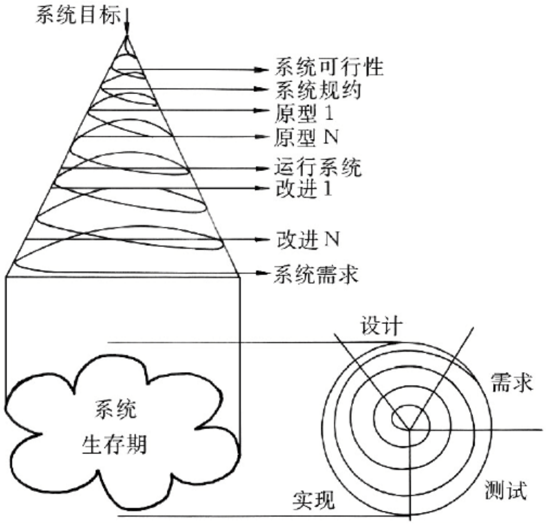


图 2　螺旋模型另一图示

如果对所开发项目的需求已有了较好的理解或较大的把握，无须开发原型，便可采用普通的瀑布模型。这在螺旋模型中可认为是单圈螺线。与此相反，如果对所开发项目的需求理解较差，需要开发原型，甚至需要不止一个原型的帮助，那就要经历多圈螺线。在这种情况下，外圈的开发包含了更多的活动。也可能某些开发采用了不同的模型。

螺旋模型适合于大型软件的开发，它是颇为实际的方法，它吸收了 T. Gilb 提出的软件工程“演化”概念。使得开发人员和客户对每个演化层出现的风险均有所了解，并继而作出反应。

和其他模型相比螺旋模型的优越性较为明显，但要求许多客户接受和相信演化方法并不容易。本模型的使用需要具有相当丰富的风险评估经验和专门知识。如果项目风险较大，又未能及时发现，势必造成重大损失。此外，螺旋模型是出现较晚的新模型，远不如瀑布模型普及，要让广大软件人员和用户接受，还有待于更多的实践。

参考文献

Boehm B. A spiral model for software dervelopment and enhancement. Computer, 1988,21(5):61-72

（郑人杰）

M

Mading-luofu leixing lilun

马丁洛夫类型理论（Martin-Löf's type theory） 瑞典逻辑学家 P. Martin-Löf 于 1971 年提出的一个类型论形式系统。又称直觉主义类型论或构造类型论。目的在于为 E. Bishop 构造数学提供全面的形式化。

E. Bishop 及其追随者的成果使构造数学得到发展。因此，为这样的构造数学作形式化是极其重要的。这理论的背景思想是基于命题作为类型观点，一命题被解释为一类型，其居元代表该命题的证明。于是谓词逻辑被解释于该类型论中。

P. Martin-Löf 原旨在将类型论构成一个框架，用于解释其他理论，从而类型论的典范化证明直接导出被解释系统的典范化证明。在该理论中，选择公理和一阶海廷算术被巧妙地表示。该类型论有多种表述，各有不同性质和特点，且具有很强的表达能力。例如（居元 a: 类型 A）可分别解释为（元素: 集合）、（证明: 命题）、（程序: 规约）、（解: 问题）。

例如马丁洛夫理论中的规则：

$$\frac{x:A\vdash b:B}{\lambda x:A.b:A\to B}\qquad\frac{m:A\to B,n:A}{mn:B}$$

按照"命题作为类型，证明作为居元"的观点，可分别表示逻辑规则

$$\frac{A\vdash B}{A\supset B}\qquad\frac{A\supset B,A}{B}$$

马丁洛夫类型论是一个规则形式系统，用规则来刻画类型及其行为，有 4 种所谓"判断"：①A type（A 为类型）；②a: A（a 有类型 A）；③$a=b\in A$（类型 A 中居元 a,b 相等）；④$A=B$（类型 A,B 相等）。该类型论中定义类型（如自然数类型）以及类型的运算（如 Π,Σ）。在引入类型的同时，类型的典范对象及其计算规则也被引入。用自然推理的式样给出这些规则，且从以下三点论证这些规则：①判断形式的语义解释；②类型定义；③典范对象的计算规则。对于每个类型、类型运算有规则组（4 条）：①形成规则（说明可由其他类型形成的类型的形式、记号）；②引入规则（说明什么是该类型的典范对象）；③消去规则（说明如何在该类型上定义函数，规定引入规则引入的居元才是典范项）；④等式规则（说明函数作用于典范对象的计算行为、外延性质）。相等概念有内涵、外延之分，也有内外之分，内相等即通常的"同一"，外相等是定义相等。在引入归约规则后，定义相等是此归约关系的构成和等价闭包。相仿地，也可由判断相等规则来定义。外相等也可以内化，可定义恒等类型（包括内涵的、外延的两类）。内涵恒等类型反映了定义相等，外延恒等类型也力图等同于定义相等，但破坏了内外差别，因此尚有争议。具有内涵定义相等和内涵恒等类型的马丁洛夫类型论是可判定的，具有外延恒等类型的是不可判定的。介乎两者中间的，具有外延定义相等但不具恒等类型的可判定性未知。

近年来，计算机科学家将马丁洛夫类型论作为程序构造的形式理论。在同一个形式系统中能同时表示规约和程序，且由证明规则从规约导出一个正确程序，并给出了验证程序性质的逻辑框架。函数式程序语言如 ML 就符合类型论思想。但是尚未进入程序规约的实用阶段。在构造数学方面，构造集合论的类型解释是很有吸引力的，尤其是否存在这样一个可判定的类型论至今尚未解决。目前已经给出一个反良基公理的马丁洛夫类型论解释，可称是美妙的结果。

20 世纪 90 年代已有不少人涉足马丁洛夫类型理论的语义研究，还有人从事类型理论的扩展以及计算机实现。

参考文献

1. Aczel P. The type theoretic interpretation of constructive set theory: inductive definitions. In: Barcan M R, Dorn G J W, Weingartner P, eds. Logic, Methodology, and Philosophy of Science. Ⅶ. North-Holland, 1986

2. Martin-Löf P. Intrutionistic type theory. Bibliopolis, Napels, 1984 （陆汝占）

mafen fuyong

码分复用（code division multiplexing, CDM） 利用编码技术，允许多个终端用户利用整个频段发送信号，在接收端分离出原始信号的一种

复用方法。传统的复用方法,如**频分复用**,总是将整个频段切割成若干小的频段,分给各用户使用。CDM 完全不同,它不作这样的分割,而是每个用户可以使用整个带宽。

有人做了一个形象的比喻:在一个大房间里面,用中文、英文、德文、西班牙文等多国语言同时广播,房间里面的人来自中国、英国、德国和西班牙等国,结果是中国人听到了中文广播,英国人听到了英语广播,德国人听到了德语广播,西班牙人只听到西班牙语广播。CDM 原理就如此,所有的工作站都同时发送信号,所有的信号都叠加到一起, 接收方只接收想接收的那个工作站的信号。具体来说,CDM 的工作原理如下:

(1) 每个终端用户被分配一个唯一的 m 位的码片序列,码片序列间是正交的;

(2) 当一个用户要发一个比特“1”的时候,就将自己的 m 位码片序列发送出去;

(3) 当一个用户要发送一个比特“0”的时候,就将自己的 m 位码片序列的补码发送出去;

(4) 所有的用户可以同时发送数据,即所有用户发出的码片序列按位进行线性叠加;

(5) 接收方用收到的叠加信号和发送方的 m 位代码进行点积运算,即可提取出发送方发出的原始信号。

为了说清楚上述 CDM 的工作原理,看这样一个例子:假设系统中有 4 个工作站(终端用户)A、B、C 和 D,每个工作站被分配了唯一的一个 8 位($m=8$)码片序列,如表 1 所示。

表 1　工作站的时间片和它发送的比特

工作站	码片序列(8 位)	发送“1”时	发送“0”时
A	−1 −1 −1 1 1 −1 1 1	−1 −1 −111 −111	111 −1 −11 −1 −1
B	−1 −1 1 −1 1 1 1 −1	−1 −11 −1111 −1	11 −11 −1 −1 −11
C	−1 1 −1 1 1 1 −1 −1	−11 −1111 −1 −1	1 −11 −1 −1 −111
D	−1 1 −1 −1 −1 −1 1 −1	−11 −1 −1 −1 −11 −1	1 −11111 −11

现在有 4 个传输实例,4 个工作站的发送情况和复用信号的情况如表 2 所示,其中,“ – ”代表该工作站不发送,“1”代表该工作站发送 1,即发出原始代码,“0”代表该工作站发送 0,即发出反码。

表 2　各工作站的发送及 CDM 复用信号

工作站发送的比特				复用信号
A	B	C	D	
—	1	1	–	$S1=B+C=(-2,0,0,0,2,2,0,-2)$
1	0	1	–	$S2=A+\bar{B}+C=(-1,1,-3,3,1,-1,-1,1)$
1	1	1	1	$S3=A+B+C+D=(-4,0,-2,0,+2,0,+2,-2)$
1	1	0	1	$S4=A+B+\bar{C}+D=(-2,-2,0,-2,0,-2,4,0)$

为了恢复出信号,只需要将复用信号和工作站的代码做点积,根据计算结果判断出工作站发送的是什么。比如,要从 S2 恢复出 A、B、C 和 D 工作站发送的比特值,只需要将 S2 分别和工作站的代码做点积如下

$S2 \cdot A=(1-1+3+3+1+1-1+1)/8=1$

$S2 \cdot B=(1-1-3-3+1-1-1-1)/8=-1$

$S2 \cdot C=(1+1+3+3+1-1+1-1)/8=1$

$S2 \cdot D=(1+1+3-3-1+1-1-1)/8=0$

点积的结果值如果是“1”,表明该工作站发送了比特值“1”;点积的结果值如果是“ −1”,表明该工作站发送了比特值“0”;点积的结果值如果是“0”,表明该工作站沉默,没有发送任何比特,这如上述结果显示的那样。

需要注意的是:

(1) CDM 适合用于无线通信系统,在真实的系统中,工作站的码片序列的位数远不止 8 位,通常是 64 位或 128 位;

(2) 工作站的码片序列之间必须是两两正交的,这可以通过 Walsh 编码来产生;

(3) CDM 是一种扩频技术,如果每秒要发送 b 位,就需要发送 mb 位,发送的信息量是原来的 m 倍。如果有 1 MHz 的频段用于 100 个工作站,采用频分复用的话,每个工作站可用频段 10 kHz,速率可达 10 kbps(假设 1 Hz 传送 1 位);如果使用 CDM 复用技术,同样的条件,每个工作站可使用完全的

1 MHz 的频段,即每秒 1 M 个码片序列,如果每“位”所用的码片序列是 100 个,传输速率跟采用频分复用相当,如果每位所用的码片序列少于 100 个,则传输速率可高过采用频分复用。

参考文献

1. Tanenbaum A S, Wetherall D J. 计算机网络. 北京:清华大学出版社, 2012

2. 纪越峰. 现代通信技术. 北京:北京邮电大学出版社, 2010

3. 张辉, 曹丽娜. 现代通信原理与技术. 西安:西安电子科技大学出版社, 2008

(袁华)

maichong ouhe shenjing wangluo

脉冲耦合神经网络(pulse-coupled neural network, PCNN) 一类神经元的不同输入之间不仅有加耦合的关系,而且有乘耦合的关系的**人工神经网络**。加耦合反映神经元信号通过突触馈接,乘耦合反映神经元信号通过离子通道连接。

1990 年,R. Eckhorn 根据猫的大脑皮层同步脉冲发放现象,提出了展示脉冲发放现象的连接域(Linking field)模型,为研究脉冲神经网络中的动态同步振荡活动提供了一个简单有效的方法。通过对 Eckhorn 提出的模型进行改进,就形成脉冲耦合神经网络(PCNN)模型。1994 年,J. L. Johnson 发表论文,阐述了 PCNN 的周期波动现象及在图像处理中具有旋转、可伸缩、扭曲、强度不变性。

图 1 给出了脉冲耦合神经网络神经元模型。该模型离散数学迭代方程如下:

$$F_{ij}(n) = e^{-\alpha_F}F_{ij}(n-1) + V_F\sum_{kl}M_{ijkl}Y_{kl}(n-1) + S_{ij} \tag{1}$$

$$L_{ij}(n) = e^{-\alpha_L}L_{ij}(n-1) + V_L\sum_{kj}W_{ijkl}Y_{kl}(n-1) \tag{2}$$

$$U_{ij}(n) = F_{ij}(n)(1+\beta L_{ij}(n)) \tag{3}$$

$$Y_{ij}(n) = \begin{cases}1, & U_{ij}(n) > E_{ij}(n-1) \\ 0, & \text{其他}\end{cases} \tag{4}$$

$$E_{ij}(n) = e^{-\alpha_E}E_{ij}(n-1) + V_EY_{ij}(n) \tag{5}$$

其中,内部活动项的耦合系数为 β,神经元强制激发的外部激励为 S_{ij},反馈输入域中放大系数和衰减时间常数分别为 V_F 和 α_F,耦合连接域 L 的放大系数和衰减时间常数分别为 V_L 和 α_L,动态门限 E 的放大系数和衰减时间常数分别为 V_E 和 α_E,权值矩阵 $\boldsymbol{M}_{ijkl}$ 和 $\boldsymbol{W}_{ijkl}$ 分别为反馈输入域和耦合连接域的连接矩阵。

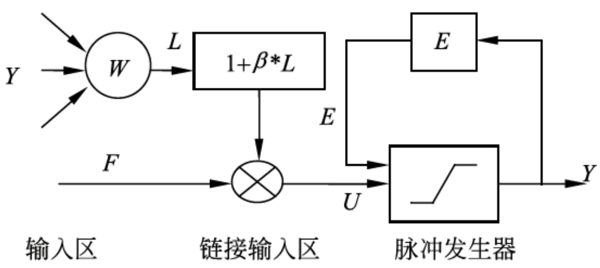


图 1 脉冲耦合神经元模型

脉冲耦合神经网络的输出序列脉冲富含外部激励的特征信息,将序列脉冲转化成一维特征序列,可较好标识外部激励,这使脉冲耦合神经网络适用于特征提取、目标识别、纹理检索和分类。另外,如果让全部神经元一个周期内发放脉冲,并以矩阵形式记录每个神经元首次发放脉冲的时刻,则该矩阵能反映出外部激励本身的同步性,可用此矩阵进行图像中噪声平滑滤波,也可用此矩阵作为脉冲耦合神经网络输出进行图像增强,研究发现其处理机理符合人眼视觉特性。脉冲耦合神经网络非常适合二维图像信息的处理,特别是在图像分割、图像平滑、噪声抑制、图像增强、特征提出、图像融合和图像检索等应用方面展现出良好性能。同步脉冲发放特性使该神经网络适用于图像分割和噪声鉴别。

近十多年来,对脉冲耦合神经网络理论和应用的相关研究深入广泛,提出了多种简化模型,包括交叉皮层模型和脉冲发放皮层模型、图像分割的脉冲耦合神经网络自动模型、图像融合的 m 通道脉冲耦合神经网络模型和单位连接脉冲耦合神经网络模型等,以改善脉冲耦合神经网络的性能,提高网络训练效率。

参考文献

1. Eckhorn R, et al. Feature linking via synchronization among distributed assemblies: simulation of results from cat cortex. Neural Computation, 1990, 2(3): 293-307

2. Johnson J L. Padgett M L. PCNN models and applications. IEEE Trans. on Neural Networks, 1999, 10(5): 480-498

3. 马义德,李廉,绽绲,王兆滨. 脉冲耦合神经网络与数字图像处理. 北京:科学出版社,2008

(马义德)

meiti chuliqi

媒体处理器(media processor) 指面向多媒体数据处理应用定制和优化的处理器。媒体处理器

主要用于音视频数据的实时流式处理。与用于计算机显示的图形处理单元(GPU)不同,媒体处理器面向的主要是数字电视以及机顶盒这样的系统,其具体应用还包括各种高清数字视频设备、DVD播放机、数字照相机、数字摄像机、视频会议系统、MPEG编码和解码系统、3D编码和解码系统等。媒体处理器可分为三种:用作媒体处理的通用型微处理器,功能固化的媒体处理器以及可编程媒体处理器。起初流媒体数据类型的处理都是采用固化的特定集成电路来实现,其缺点是当流媒体处理的流程或者数据格式有任何更改的时候,这种固化的处理方式都很难进行相应更新;而可编程的媒体处理器则没有这个问题,也避免了频繁重新设计硬件的工作。

多媒体处理,特别是数字视频信号处理在20世纪80年代末期开始广泛地得到推广。最初的数字化的视频信号大多采用通用**微处理器**进行处理。由于这些处理需要进行大量的数字信号卷积和变换操作,在性能较高的应用中也广泛采用通用的数字信号处理器(DSP)。但因受到当时通用微处理器和数字信号处理器运算速度的限制,很难达到实时处理的要求。在此之后,随着JPEG、MPEG静态和动态图像压缩标准(参见**图像的压缩编码**)以及H.263、H.324等视频会议标准的出现,伴随消费者对高质量多媒体应用的需求不断增强,多种类型的媒体处理器都得到了不同程度的发展:

(1)通用微处理器通过逐步增加针对多媒体处理的扩充指令集,使其实际上已成为应用最广泛的媒体处理器。比如Intel及AMD在1996推出的MMX指令集,1998年推出的3D-Now指令集,1999年推出的SSE指令集等。其中SSE指令集在2000年、2004年以及2007年又分别进化为SSE2、SSE3以及SSE4。嵌入式通用处理器方面,ARM所采用的Neon信号处理扩展集也属于此类。这些指令集可以使得诸如MPEG回放等操作所需的指令数量减少,这样可在通用型计算机上实现更快的媒体处理。随着体系结构的发展,一些诸如MPEG4压缩等原先无法承受的媒体处理任务,在通用微处理器上也已能达到实时处理的性能。通用型计算设备处理图像、音频、视频信号的性能的不断提高,也促进了MP3、智能手机等新的多媒体应用的出现和普及。

(2)专用的媒体处理电路随着MPEG等多媒体技术的标准化而成为可能。这些电路通常只能满足特定的应用需求(如MPEG编解码等)。针对具有特定功能的专业视频设备,或具有广大市场的消费产品:如VCD播放机、DVD播放机、视频电话终端、高清摄像机等,专用媒体处理器通过硬件的方法把特定的多媒体处理功能实现在一片或多片超大规模集成电路上。此类媒体处理器具有性能高、价格便宜、省电、系统设计简单等优点;其缺点是功能固定,不能用于其他媒体处理。

(3)可编程媒体处理器结合了通用微处理器和专用媒体处理器的优点,既可达到很高的处理性能,又有可编程和可适应不同媒体处理要求的特点。通常这一类媒体处理器采用类似于可编程数字信号处理器的高度并行体系结构。处理器内部的不同专用和通用的处理部件可由机器的指令集控制,因而可以通过采用编程的方式来实现特定应用所需的多样的媒体处理算法。由于此类处理器可高速地执行媒体处理中大量的基本操作和具有高度的数据并行性,因而可达到普通处理器所不能及的处理性能。

硬件结构

常见的可编程媒体处理器体系结构可分为两类:**数字信号处理器**(DSP)体系结构和超长指令字(VLIW)体系结构。

采用DSP体系结构的产品有TI公司的TMS320C80多媒体信号处理器以及Chromatic Research的MPACT媒体处理器。这一类体系结构的特点是采用多个并行信号处理器(DSP)核来提高媒体处理器的运算速度。如前所述,视频和音频处理涉及大量的数字信号处理算法(如DCT数字余弦变换、数字卷积和滤波等),因此采用数字信号处理器来进行这些运算是一个很自然的选择。但由于单个数字信号处理器核的计算速度不能达到某些应用的实时处理要求,因此需要采用多个数字信号处理器核进行并行处理。这类数字信号处理器和普通数字信号处理器类似,通常具有高速**算术逻辑部件**(ALU)和专用的高速乘加部件以适合卷积等常用的数字信号处理运算。数字信号处理器核的寄存器堆的设计通常和其运算部件相结合:采用分布式寄存器堆。除了通用寄存器以外,还包括了每个运算器专用的寄存器(如乘法结果寄存器等)。这些专用寄存器通常包含在各个运算或功能部件内以提高数据通道访问的速度,且在指令集编码上也比较简洁。数字信号处理器核的存储系统中常采用程序和数据分离的哈佛结构。在数据空间中,通常有较大的高速片内存储器,便于存放当前操作的数据窗口以及中间结果。有些媒体处理器还有专门的外设总线,用以提高多媒体数据的输入输出速率。在媒体

处理器的多个数字信号处理器核之间通常采用高速内部总线、**共享存储器**或交叉开关等结构进行互连。并行的数字信号处理器核则采用多指令流多数据流作业的方式来进行媒体数据处理,或者采用单指令流多数据流的方式来实现数据并行作业。通常媒体处理器还包括一个控制微处理器以协调各个部件之间的操作。此外,媒体处理器还包括一些高速外设部件和 DMA 控制器,使得多媒体数据得以高效地输入和输出。

基于超长指令字(VLIW)体系结构(参见**VLIW 处理器**)的媒体处理器包括 Philips Trimedia 媒体处理器、Equator 媒体处理器以及 Princeton 大学的 VSP 研究计划等。这类体系结构与基于 DSP 的媒体处理有很多共同特点,主要的不同在于基本的处理部件不是采用数字信号处理器,而是采用超长指令字的指令级并行处理器,并结合了向量处理器的特点。由于高效地实现具有指令级并行性的应用程序,与基于多个 DSP 的媒体处理器相比,VLIW 媒体处理器更便于编程。程序员只需要用高级语言编写单一指令流的程序,利用优化编译器就可自动提取程序中的指令级并行性,并把它映射到 VLIW 处理器的相关部件上,这样就避免了并行编程的难度。

发展趋势

媒体处理器发展的基本动力在于人们对涉及图形、图像、声音等高质量的多媒体应用的不断提高的广泛要求以及多媒体算法的不断更新和改进,以上提到的三类媒体处理器都将相对独立地得到发展:基于通用微处理器的多媒体个人计算机平台将随着处理器主频的提高和体系结构的改进而得到提高和发展;专用的媒体处理器将随着高清晰度电视的普及、3D 技术的出现以及新的数字电视存储标准的成熟而继续为家电产品提供成熟低价的媒体处理方案;在多媒体算法以及数字电视和数字音乐算法发展的阶段,可编程媒体处理器将为这些算法提供高性能和方便的实现平台。

参考文献

1. Katz D J, Gentile R. 嵌入式媒体处理. 陈喆,殷福亮,景源,译. 北京:电子工业出版社,2007

2. Catthoor F, Raghavan P, Lambrechts A, et al. Ultra-low energy domain-specific instruction-set processors(Embedded Systems). London: Springer, 2010

3. 张太镒,等. 音视频多媒体处理技术与实践——基于 i. MX27 处理器. 北京:北京航空航天大学出版社,2011

(廖恒 余宏亮)

Mengtekaluo fa

蒙特卡罗法(Monte Carlo method) 以概率和统计的理论与方法为基础的一种数值计算方法。

将所求解的问题同一个概率模型相联系,用计算机实现统计模拟或抽样,从而求得问题的近似解,亦称为统计试验法或统计模拟法。蒙特卡罗方法为双重近似,一是概率模型模拟近似的数值计算,二是用伪随机数模拟真正的随机变量的样本。

蒙特卡罗法在 1945 年左右由 J. von Neumann 和 S. M. Ulam 为研制核武器的需要而首先提出来的,但远在 1777 年,有人就提出由投针实验求圆周率的方法,这应该算是最早的蒙特卡罗法。

积分计算 设 $g(x)$ 是$[0,1]$上连续函数,且 $0 \leqslant g(x) \leqslant 1$, 计算 $S = \int_0^1 g(x)\mathrm{d}x$。$S$ 就是由 $g(x)$ 及 $x = 0, x = 1$ 与 x 轴所界的图形 G 的面积。考虑在正方形$[0,1;0,1]$上独立的均匀分布随机变量 (ξ,η),则 $P((\xi,\eta) \in G) = \int_0^1 \int_0^{g(x)} \mathrm{d}y\mathrm{d}x = S$。若向正方形$[0,1;0,1]$内均匀投点 n 次,点的坐标为 (ξ_i,η_i), $i = 1,2,\cdots,n$。计算 (ξ_i,η_i) 落入 G(亦即满足 $\eta_i \leqslant g(\xi_i)$)的个数 k,由强大数定律, $P\left(\lim_{n\to\infty} \frac{k}{n} = S\right) = 1$, 于是可用 k/n 作为 S 的近似值。这里所谓向正方形投点,就是构造二维的独立的均匀分布伪随机数列,并计算 k。这里 k 是一个随机变量,是 n 次伯努利试验(投点)中,成功(点落入 G)的次数。由于 $E\left(\frac{k}{n}\right) = S$, $D\left(\frac{k}{n}\right) = \frac{S(1-S)}{n}$,且 $\frac{k-nS}{\sqrt{D(k)}}$ 渐近于 $N(0,1)$ 分布,因此

$$P\left(\left|\frac{k}{n} - S\right| < \varepsilon\right)$$

$$\approx \Phi\left(\varepsilon\sqrt{\frac{n}{S(1-S)}}\right) - \Phi\left(-\varepsilon\sqrt{\frac{n}{S(1-S)}}\right)$$

这里 Φ 是 $N(0,1)$ 分布函数。查表可求得 $\Phi(t_\alpha) - \Phi(-t_\alpha) = \alpha$,于是得 $n \approx t_\alpha^2 S(1-S)/\varepsilon^2$。但实际上并不知道 S 的值,因此必须首先估计 S。可用试算或其他办法得到 S 的估计值 $\hat{S}$,代入上式求得 n,再做投点试验,得到 S 满足精度的估计。从上述 n 与 ε 的关系可以看出,增加 n 对结果的精度影响不大,但如果能改进抽样方法,降低方差 $D\left(\frac{k}{n}\right)$ $\left(= \frac{S(1-S)}{n}\right)$便可提高精度。

还可采用平均值法计算上述积分。任取[0,1]上均匀分布样本 ξ_i，则 $g(\xi_i), i=1,2,\cdots$ 也是相互独立同分布的随机变量，且 $Eg(\xi_i)=\int_0^1 g(x)\mathrm{d}x=S$。由强大数定律，可取

$$S_n=\frac{1}{n}\sum_{i=1}^{n}g(\xi_i)$$

作为 S 的估计。如 $g(x)$ 在[0,1]上平方可积，则

$$D(S_n)=\frac{1}{n}Dg(\xi_i)=\frac{1}{n}\left[\int_0^1 g^2(x)\mathrm{d}x-\left(\int_0^1 g(x)\mathrm{d}x\right)^2\right]$$

这时有近似公式

$$n\approx\frac{1}{\varepsilon^2}t_\alpha^2 D(g(\xi_i))$$

t_α 为 t 分布的临界值。由于 $Dg(\xi_i)\leqslant S(1-S)$，可见为了得到同样的精度，在相同置信度 α 下，平均值法比投点法所需的计算次数 n 要小。

上述方法不难推广到任意区间[a,b]上，而要求 $0\leqslant g(x)\leqslant 1$ 是非本质的。对于计算高维数值积分

$$S=\int_D\cdots\int g(x_1,\cdots,x_m)\mathrm{d}x_1\cdots\mathrm{d}x_m$$

其中 D 为 m 维单位区间，也可用类似的方法。比如考虑 m 维单位区间上的均匀分布随机向量 $\boldsymbol{\xi}=(\xi_1,\cdots,\xi_m)$，任取 n 个样本 $\{\xi^i, i=1,2,\cdots,n\}$，则可用 $S_n=\frac{1}{n}\sum_{i=1}^{n}g(\xi^i)$ 来近似 S。

对于低维积分，一般数值求积法的精度比较高，但对于四维以上的积分，蒙特卡罗法就更具优势。在实际应用中，可以将这两种方法结合起来，尽可能用分析方法算出对某些变元的积分，以降低其维数，然后再用统计试验法求出剩下变元的积分。

蒙特卡罗法还可用于解线性方程组、矩阵求逆、常微分方程边值问题求解、偏微分方程求解、非齐次积分方程求解、特征值计算和最优化计算等。

参考文献

1. Shreider Yu A. Method of statistical testing (Monte Carlo method). Amsterdam/London/New York: Elsevier Publishing Company, 1964

2. 中山大学数力系. 概率论及数理统计(下册). 北京：人民教育出版社，1980　(金治明)

Mengtekaluo fangzhen

蒙特卡罗仿真(Monte Carlo simulation)　以概率统计理论为指导，使用随机数(或伪随机数)来解决某种随机性或确定性问题的一种仿真方法。也称为随机仿真(random simulation)或随机抽样(random sampling)。

蒙特卡罗仿真的基本思想很早以前就被人们所发现和利用。早在17世纪，人们就知道用事件发生的“频率”来决定事件的“概率”。19世纪人们用投针试验的方法来决定圆周率 π。20世纪40年代**电子计算机**的出现，特别是近年来高速电子计算机的出现，使得用数学方法在计算机上大量、快速地模拟这样的试验成为可能。蒙特卡罗仿真的名称源于美国在第一次世界大战研制原子弹的“曼哈顿计划”。该计划的主持人之一、数学家冯·诺依曼用赌城——摩纳哥的Monte Carlo——来命名这种方法。

蒙特卡罗仿真的一般步骤：

(1) 根据提出的问题构造一个简单、适用的概率模型或随机模型，使问题的解对应于该模型中随机变量的某些特征(如概率、均值和方差等)，所构造的模型在主要特征参量方面要与实际问题或系统相一致。

(2) 根据模型中各个随机变量的分布，在计算机上产生随机变量，实现一次仿真过程所需的足够数量的随机数。通常先产生均匀分布的随机数，然后生成服从某一分布的随机变量。

(3) 根据概率模型的特点和随机变量的分布特性，设计和选取合适的抽样方法，并对每个随机变量进行抽样(包括直接抽样、分层抽样、相关抽样、重要抽样等)。

(4) 按照所建立的模型进行仿真试验、计算，求出问题的随机解。

(5) 统计分析仿真试验结果，给出问题的概率解以及解的精度估计。

蒙特卡罗仿真属于静态仿真，即模型中一般不考虑系统状态随时间变化的动态行为过程。这些问题可分为两类：第一类是确定性的数学问题，如计算多重积分、解线性代数方程组等；第二类是随机性问题，如原子核物理问题，运筹学中的库存问题，随机服务系统中的排队问题，动物的生态竞争和传染病的蔓延问题等。

蒙特卡罗仿真的特点是程序结构简单，便于编制和调试，其仿真样本数可以依据仿真需要的精度灵活地选取；弱点是收敛速度慢，对于大系统适用性差。因此，它是以高容量和高速度的计算机为前提条件的。将其他方法与蒙特卡罗方法联合起来使用，可克服这种局限性。

参考文献

徐钟济. 蒙特卡罗. 北京：科学出版社，1985

（王威　肖田元）

mimaxue

密码学（cryptography） 以研究数据保密为目的，对存储或传输的信息进行秘密的变换以防止被第三者窃取的技术。被变换的信息称为**明文**，它可以是一段有意义的文字或者数据；变换过后的形式称为**密文**，密文应该是一串杂乱排列的数据，从字面上没有任何含义。从明文到密文的变换过程称为加密。变换本身是一个以加密密钥 k 为参数的函数，记作 $E_k(P)$。密文经过通信信道的传输到达目的地后需要还原成有意义的明文才能被通信接收方理解。将密文 C 还原为明文 P 的变换过程称为解密或者脱密。该变换是以解密密钥 k' 为参数的函数，记作 $D_{k'}(C)$。密码学的加密解密模型如图 1 所示。

在传统密码体制中，加密和解密采用的是同一密钥，即 $k=k'$，并且 $D_{k'}(E_k(P))=P$，称为**对称密钥密码系统**，也称**私钥密码系统**。私钥密码系统中使用的密钥常称作**传统密钥**或**传统密码**。现代密码体制中加密和解密采用不同的密钥，称为**非对称密钥密码系统**，每个通信方均需要有 k 和 k' 两个密钥。在进行保密通信时通常将加密密钥 k 公开，称为**公钥**，而将解密密钥 k' 保密，称为**私钥**，所以也称为**公钥密码系统**。传统密码系统中最常见的算法有数据加密标准（DES）、国际数据加密算法（IDEA）等。DES 算法是 IBM 开发的并于 1977 年被美国政府采纳为非机密信息的加密标准。它的原始形式已经在 1995 年被人攻破，但其修改后的形式仍然是有效的。IDEA 是 Lai 和 Massey 提出的，目前还没有发现有效的攻击方法。

密码学研究的内容包含两部分：一是加密算法的设计和研究，另外一部分是密码分析，也就是密码破译技术。在密码学中，如果进行密码分析的攻击者对密码通信进行攻击时，仅能够被动地监听通信信道上所有信息，就被称为被动攻击；如果攻击者还能够对通信信道上传输的消息进行截取、修改甚至主动发送信息，则称之为主动攻击。攻击者与报文接收方的区别在于是否知道解密密钥。由于攻击者不知道解密密钥，因此很难将密文脱密还原为明文。

公钥方案较对称密钥方案处理速度慢，因此，通常把公钥与对称密钥技术结合起来实现最佳性能。即用**公钥密码技术**在通信双方之间传送对称密钥，而用**私钥密码技术**来对实际传输的数据加密、解密。另外，公钥加密也用来对私钥进行加密。

因为在现代密码学研究中对于加密和解密算法一般都是公开的，所以对于攻击者来说只要知道解密密钥就能够破译密文。因此密钥设计被称为密码学研究的核心，密钥保护被视为防御攻击的重点。对于密钥分析来说，对其进行穷举猜测攻击是任何密码系统都无法避免的，但是当密钥长度足够大并且足够随机就会使得穷举猜测在实际上变得不可能。例如，密钥长度为 256 位的加密算法，密钥空间为 2^{256}，对应为 1077 量级。如果一台计算机每秒可以对密钥空间进行一亿次搜索，那么全部搜索一遍事件所需的时间以年为单位将大于 10^{62}。如果密钥空间小或者分布具有一定可预见性，那么攻击者就可能利用相关知识大大缩小搜索空间，从而破译密文。

参考文献

1. 胡道元. 计算机网络（高级）. 北京：清华大学出版社，1999

2. 卢开澄. 计算机密码学. 北京：清华大学出版社，1998

（胡道元）

mianxiang duixiang chengxu sheji

面向对象程序设计（object-oriented programming） 设计与构作面向对象程序的方法与

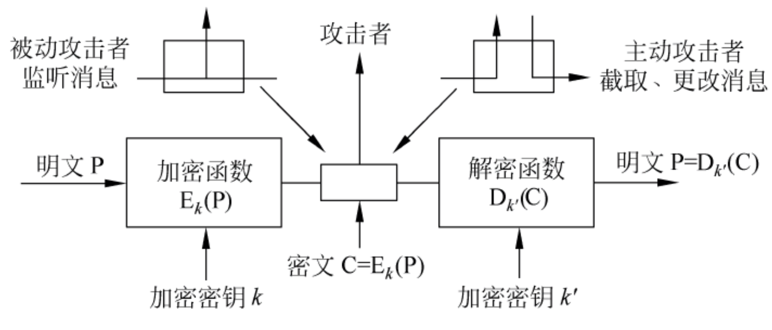


图 1　密码学模型

过程。面向对象程序设计以**对象**为核心,对象是程序运行时刻的基本成分。面向对象程序设计语言中提供了**类**、**继承**等设施。面向对象程序设计即为设计类及由类构作程序的方法与过程,用计算机对象模拟现实世界对象。

现实世界中的系统由一组相互联系、协同作用的基本构件组成,这些基本构件称作客观对象。当用面向对象的方法进行计算机模拟时,在计算机系统中也应有其对应成分,即计算机对象,简称对象。面向对象程序设计方法就是面向这样的对象构作程序乃至系统。

结构化程序设计侧重于功能抽象,它将解决问题的过程视为一个处理过程,每个模块都是一个处理单位。面向对象程序设计综合了功能抽象和数据抽象,它将解决问题的过程视为一个分类演绎过程,对象是数据和操作的封装。在结构化程序设计中,过程为一独立实体,显式地为使用者所见;而在面向对象程序设计中,操作是对象私有的,只能通过消息传递来引用。如果参数相同,则过程调用的结果相同,而同一消息的多次发送可能产生不同的结果,取决于对象的当前状态。

面向对象程序设计中的基本概念有对象、类、继承、多态和动态绑定。

参考文献

1. 徐家福,等. 对象式程序设计语言. 南京:南京大学出版社,1992

2. Shriver B, Wegner P. Research directions in object-oriented programming. MIT Press,1987

(张家重　王志坚)

mianxiang duixiang fangfa

面向对象方法(object-oriented method)　一种把面向对象的思想运用于软件开发过程中,指导开发活动的系统方法。简称 OO 方法,是建立在"对象"概念(对象、类和继承)基础上的方法学。**对象**是由数据和容许的操作组成的封装体,与客观实体有直接的对应关系。一个对象类定义了具有相似性质(属性)的一组对象。而**继承性**是对具有层次关系的类的属性和操作进行共享的一种方式。所谓**面向对象**就是基于对象概念,以对象为中心,以类和继承为构造机制,来认识、理解、刻画客观世界和设计、构建相应的软件系统。

面向对象的方法起源于面向对象的编程语言。20 世纪 60 年代后期 Simula-67 语言中出现了类和对象的概念,类作为语言机制用来封装数据和相关操作。70 年代前期,A. Kay 在 Xerox 公司设计出了 Smalltalk 语言,奠定了面向对象程序设计的基础,1980 年 Xerox 公司推出了商品化的 Smalltalk-80,标志着面向对象的程序设计已进入实用阶段。进入 80 年代相继出现了一系列面向对象的编程语言。如:C++,Object-C,Clos,Eiffel 等。自 80 年代中期到 90 年代,OO 的研究重点已经从语言转移到设计方法学方面,尽管还不成熟,但已陆续提出了一些面向对象的开发方法和设计技术。其中具有代表性的工作有:B. Henderson-Sellers 和 J. M. Edwards 提出的面向对象软件生存周期的"喷泉"模型及面向对象系统开发的七点框架方法;G. Booch 提出的面向对象开发方法学;P. Coad 和 E. Yourdon 提出的面向对象分析(OOA)和面向对象设计(OOD),J. Rumbaugh 等提出的 OMT 方法学等。这些方法的提出,标志着面向对象方法逐步发展成为一类完整的方法学和系统化的技术体系。而有关抽象数据类型的基础研究为面向对象开发方法提供了初步的理论。

面向对象方法的具体实施步骤如下:

面向对象分析　从问题陈述入手,分析和构造所关心的现实世界问题域的模型,并用相应的符号系统表示。模型必须简洁、明确地抽象目标系统必须做的事,而不是如何做。分析步骤为:①确定问题域。包括:定义论域,选择论域,根据需要细化和增加论域。②区分类和对象。包括定义对象,定义类,命名。③区分整体对象及其组成部分,确定类的关系及结构。包括:一般-具体结构,整体-部分结构,多重结构。④定义属性。包括确定属性,安排属性。确定实例联结。⑤定义服务。包括确定对象状态,确定所需服务,确定消息联结。⑥确定附加的系统约束。

面向对象设计　因为 OO 方法设计的软件系统结构本质上是并行的,并且突出相对稳定的数据结构,因此 OO 方法与传统的以功能分解为主的方法学的设计内容和步骤有所不同。具体设计步骤如下:①应用面向对象分析对用其他方法得到的系统分析的结果进行改进和完善。②设计交互过程和用户接口。包括:描述用户及任务,并根据需要分成子系统;把交互作用设计成类;设计命令层次;设计交互作用过程及接口,并用相应符号系统表示。③设计任务管理。根据前一步骤确定是否需要多重任务;确定并发性;确定以何种方式驱动任务;设计子系统及任务之间的协调与通信方式;确定优先级。

④设计全局资源协调,确定边界条件,确定任务或子系统的软、硬件分配。⑤设计各个类的存储,数据格式;设计实现类所需的算法;将属性和服务加入到各个类的存储对象中;设计对象库或数据库,前四个步骤叫系统设计,最后一步叫对象设计。

面向对象实现 设计阶段设计的对象和关联最终都必须用具体的编程语言或数据库实现。使用 OO 语言来实现 OO 设计相对来说比较容易,因为语言的构造与设计的构造是相似的,OO 语言支持对象、运行多态性和继承。使用非 OO 语言需要特别注意和规定保留程序的 OO 的结构,OO 概念可以映射到非 OO 语言结构中,这只是一个表达方式的问题,不是语言的能力问题,因为编程语言最终要转换为机器语言,但 OO 语言良好的风格尤为突出。

在传统的面向功能的方法学中,强调的是确定和分解系统功能,这种作法虽然是目标的最直接的实现方式,但由于功能是软件系统中最不稳定、最容易变化的方面,因而获得的程序往往难于维护和扩充,OO 方法首先强调认识来自应用域的对象,然后围绕对象设置属性和操作。用 OO 方法开发的软件,其结构源于客观世界稳定的对象结构,与传统软件相比,软件本身的内部结构发生了质的变化,易重用性和易扩充性都得到提高。围绕对象来组织软件和进行软件设计,可将现实世界模型直接自然地映射到软件结构中,可望从根本上解决软件的复杂性问题。并且基于这种新的软件结构,可使软件通过构造的方法自动生成,从而提高软件的生产率和质量。由于软件是建立在应用域自身基础的基本构架之上,不是单个问题的指定功能需求,因而能更好地支持软件需求的演化。

总之,面向对象的开发方法不仅为人们提供了较好的开发风范,而且在提高软件的生产率,可靠性、易重用性、易维护性等方面有明显的效果,已成为当代计算机界最为关注的一种开发方法。

参考文献

1. Coleman D. Object-oriented development: the fusion method. New York: Prentice Hall,1994

2. Rumbaugh J. Object-oriented modeling and design. New York: Prentice Hall,1991

(郝克刚　鱼滨)

mianxiang duixiang shujuku

面向对象数据库(object-oriented database)

支持面向对象数据模型的数据库。面向对象数据库简称 OODB。面向对象数据库管理系统简称 OODBMS。

用户使用面向对象数据库有两个主要原因:①**关系数据库**在用于管理复杂数据时表达能力不足;②许多应用软件是用面向对象编程语言编写的,用来转换程序中的对象和关系数据库元组的代码很烦冗,执行也很耗时。

面向对象数据模型的基本内容如下:

(1) 现实世界中的任何事物都可以被建模为**对象**。每个对象具有一个统一标识,称作对象标识。

(2) 对象是其状态和行为的封装,其中状态是对象属性值的集合,行为是变更对象状态的方法集合。

(3) 具有相同属性和方法的对象的全体构成了**类**,类中的对象称为类的实例。

(4) 类的属性的定义域也可以是类,从而构成了类的复合。类具有继承性,一个类可以继承另一个类的属性与方法,该类称为被继承的类的子类,被继承的类称为该类的**超类**。类与类之间的复合与继承关系形成了一个有向无环图,称为类层次。

(5) 对象是被封装起来的,它的状态和行为在对象外部不可见,从外部只能通过对象显式定义的消息传递对对象进行操作。

与传统数据库一样,面向对象数据库对数据的操纵包括数据查询、增加、删除、修改等。面向对象数据库也具有并发控制、故障恢复、存储管理等功能。

面向对象数据库不仅能支持传统数据库应用,也能支持非传统领域的应用,包括 CAD/CAM、OA、CIMS、GIS 以及图形、图像等多媒体领域、工程领域和数据集成等领域。

面向对象数据库的研究始于 1985 年。著名的研究项目包括美国布朗大学的 Encore-Observer、美国威斯康星大学的 EXODUS、美国惠普公司的 IRIS、美国贝尔实验室的 ODE、MCC 的 ORION 和德州仪器公司的 Zeitgeist。

面向对象数据库产品较多。较著名的有 Object Store、O2、ONTOS 等,其中 Object Store 是对面向对象程序设计语言 C++ 的持久性扩充,它主要流行于美国;ONTOS 与 Object Store 类似;O2 是一个不依赖于其他软件工具而独立存在的面向对象数据库,它有一个类 SQL 语言,主要流行于西欧各国。

面向对象数据库的标准化程度不够,已有的标准有 ANSI 1991 年的“OODBTG Final Report”,它给

出了面向对象数据库的参考模型和实现的标准化建议。有关面向对象数据库的终端查询语言标准至今尚未统一,可供参考的标准有 ANSI 的 SQL-和 OMG 的 ODMG。

参考文献

1. Kim W. Introduction to object oriented database system. MIT Press, 1990

2. Kemper A. Object-oriented database management. Prentic Hall Inc. , 1994

(李建中　徐洁磐　邹兆年)

mianxiang duixiang yuyan

面向对象语言(object-oriented language)　用于描述面向对象程序的**程序设计语言**。面向对象程序设计以对象为核心,对象是程序运行时刻的基本成分。语言中提供了**类**、**继承**等设施。

面向对象语言借鉴了 20 世纪 50 年代的人工智能语言 LISP,引入了动态**绑定**和交互式开发环境的思想;始于 60 年代的离散事件模拟语言 SIMULA 67,引入了类的概念和继承;成形于 70 年代的 Smalltalk。面向对象语言的发展有两大方向:一是纯面向对象语言,如 Smalltalk,EIFFEL 等;另一是混合型面向对象语言,即在过程式语言或其他语言中加入类、继承等成分,如C ++ ,Objec-C 等。

面向对象语言刻画客观系统较为自然,便于软件扩充与复用。其主要特点有四:①识认性,系统中的基本构件可识认为一组可识别的离散对象;②类别性,系统具有相同数据结构与行为的所有对象可组成一类;③多态性,对象具有唯一的静态类型和多个可能的动态类型;④继承性,在基于层次关系的不同类中共享数据和操作。其中,前三者为基础,继承是特色,四者(有时再加上动态绑定)结合使用,体现出面向对象语言的表达能力。

一般认为,较典型的面向对象语言有:①SIMULA 67,支持单继承和一定含义的多态和部分动态绑定;②Smalltalk 支持单继承、多态和动态绑定;③EIFFEL,支持多继承、多态和动态绑定;④C ++ ,支持多继承、多态和部分动态绑定。四种语言涉及概念的含义虽基本相同,但所用术语有别。

参考文献

1. 徐家福,等. 对象式程序设计语言. 南京:南京大学出版社,1992

2. Special Issue on object-oriented design. CACM, 1990,33(9)

(张家重　王志坚)

mianxiang duixiang zhongjianjian

面向对象中间件(object-oriented middleware)　基于**对象请求代理**机制的**分布计算中间件**。面向对象中间件支持分布异构环境下对象之间通信和交互,实现分布对象之间互联互通互操作,并支持面向对象的分布式系统的开发、部署、运行与管理,是目前应用较广的主流中间件之一。

面向对象中间件通常由对象请求代理机制、公共服务和相关工具组成。对象请求代理机制是面向对象中间件的核心,支持分布对象之间的透明访问,是所有客户对象与服务对象连接的桥梁,因而称为软总线,如图 1 所示。该总线为分布对象间的连接提供了标准接口,换句话说,各种分布对象通过标准接口均可连接到软总线上,实现即插即用。公共服务为分布对象通过软总线实现互联互通互操作提供基本的共性服务,例如名字服务、事件服务、事务处理服务以及通告服务等。工具部分主要有开发工具、集成工具、部署工具和管理工具。

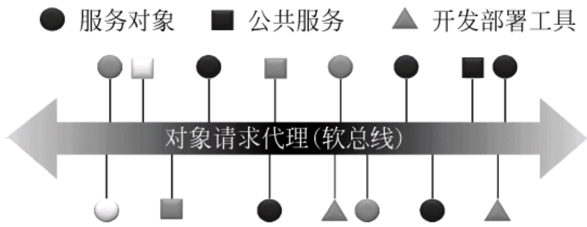


图 1　对象请求代理软件总线

CORBA、JEE 和. NET 是面向对象中间件中三个最具有代表性的技术体系。其中,CORBA 是最通用的对象请求代理体系,具有跨编程语言、软硬件平台和网络协议的特性。面向对象中间件在分布式系统的集成与资源共享中正在发挥重要作用,并已平台化,成为快速构建各种分布式应用系统的重要基础平台。

参考文献

OMG. Common Object Request Broker Architecture Specification. http://www. omg. org/technology/documents/corba_spec_catalog. htm

(史殿习　丁博)

mianxiang fangmian de chengxu sheji

面向方面的程序设计(aspect-oriented programming,AOP)　起源于程序设计中的代码散布(code scattering)和代码交织(code tangling)问题。在软件系统开发过程中,开发人员开发和维护一个

系统以满足多个需求,可以把这些需求分类为核心模块级需求和系统级需求。很多系统级需求,例如日志、验证、资源池、系统管理、性能及存储管理等,一般都会横切(crosscutting)许多核心模块。因而,在这种情况下,类不仅要处理自己的核心关注点,以便实现核心模块级需求,而且还必须实现横切关注点,以便满足相关的系统级需求。这种关注点的混合,导致了代码散布和代码交织。

代码散布　由于横切关注点本来就涉及多个模块,其相关实现也就遍布在这些模块中,如在一个使用了日志的系统里,其日志的读写问题就会影响所有进行访问和操作的模块。这导致代码散布在各处。

代码交织　软件系统中的模块可能要同时兼顾几个方面的需要。举例来说,开发者经常要同时考虑业务逻辑、性能、同步、日志和安全等问题,兼顾各方面的需要导致相应关注点的实现元素同时出现,引起代码交织。

在目前的程序设计技术下,这两种现象可以说是不可避免的,是软件复杂度的来源之一。

在1997年欧洲面向对象程序设计会议上首次提出了面向方面程序设计(aspect-oriented programming,AOP)的概念。AOP就是要能分离出那些隐含的、相互交织纠缠的系统关注点,并使之明确。使用AOP为程序员提供的新的模块化机制,横切的代码就可以局部化,程序的结构就会变得更清晰,代码也常常很简洁。作为一种编程范型,AOP的实现可以分为两部分,即AOP的语言规范和语言实现。语言规范定义了语言的基本元素和语法语义,语言实现则定义了按照这种语言规范来验证代码的合法性并通过编译转化为目标机器的可执行方式。

(1) AOP语言规范　通过建立一种语言来指定关注点织入的规则。目前常用的面向方面编程语言有许多种,其中AspectJ语言是最常用的面向方面编程语言之一。它是对Java语言的面向方面扩展。AspectJ通过向Java中引入连接点(join point)、切入点(pointcut)、通知(advice)、类型间声明(inter-type declaration)和方面(aspect)等概念,实现了AOP的核心思想,使核心关注点和横切关注点无缝集成。

(2) AOP的实现　用于将关注点(包括核心关注点和横切关注点)转化为可执行的代码,从而将横切关注点代码和主应用程序集成在一起。这种工作机制也叫作编织或织入(weaving)。AOP编织的类型是多种多样的,有编译期织入、链接期织入、载入期织入和运行期织入等。采用编织这一方式,将面向方面程序中改变主程序的横切关注点代码与原面向对象程序进行有机地混合,从而实现了面向方面程序为原面向对象程序提供的各种横切功能。

参考文献

1. Kiczales G, Lamping J, Mendhekar A, et al. Aspect-oriented programming. In: Proceedings of the 11th European Conference on Object-Oriented Programming, 1997: 220-242

2. Kiczales G, Hilsdale E, Hugunin J, et al. An overview of AspectJ. In: Proceedings of the 15th European Conference on Object-Oriented Programming, 2001: 327-353

3. Hilsdale E, Hugunin J. Advice weaving in AspectJ. In: Proceedings of 3rd International Conference on Aspect-oriented Software Development, 2004: 26-35

(赵建军)

mianxiang fuwu de ruanjian kaifa fangfa

面向服务的软件开发方法(service-oriented software development method,SOSD)　随着互连网络的快速发展,企业应用呈爆发式增长。由于业务需求的多样性和复杂性,企业之间开始寻求一种跨企业的应用集成方法,面向服务架构(service-oriented architecture,SOA)由此应运而生,它致力于制定企业应用的标准、开放接口和通信协议,并依此为基础解决跨企业应用集成问题。SOA是一种与通信协议和编程语言无关的分布式计算范型,它将互连网络上的计算资源以服务的形式进行包装、描述、发布和使用,SOA具有如下三个基本属性:

(1) 技术中立　通信、描述和发现机制与广泛接受的技术标准相兼容。

(2) 松散耦合　服务之间的依赖关系,包括时间依赖、地址依赖、协议依赖、事务依赖等都是松散的。

(3) 透明性　通过间接寻址发现服务资源。

随着面向服务架构SOA相关技术的进一步发展,逐渐形成支持面向服务的软件开发的这一新的分布式软件开发方法。

面向服务的软件开发方法SOSD通过组合可重用的服务实现软件系统的开发,通常这些服务可由

众多服务提供商提供。因此,面向服务的软件开发方法主要关注如何按需、动态地集成现有的服务,从而快速开发出满足新需求的软件系统。在 SOA 下,服务提供者和服务使用者之间的交互关系具有动态特性,服务公开性(exposure)和反射性(reflection)替代了传统的固定式系统集成,开发软件系统时是根据系统的需求进行服务装配与组合。此外,服务的松耦合改变了传统以 APIs 调用进行组件组装的紧耦合方式,系统架构师可以通过动态描述组合服务集合来创建软件系统。

广义上讲,面向服务的开发方法学包括项目规划、系统分析和设计、系统实施、系统部署和维护以及整个过程中的监控和管理等。从实践的角度说,面向服务的开发方法包含了如下几个方面:

(1) 面向服务的分析和设计(service oriented analysis and design,SOAD) 以服务为中心,根据业务需求识别服务、描述服务,并设计服务的实现。

(2) 面向服务的开发过程 结合现有开发过程,规划以服务为中心的开发过程中的角色、职责、活动和工件。

(3) SOA 的成熟度分析和迁移路线 以服务为中心,分析现有或目标系统的成熟度,并设计从现有成熟度迁移到目标成熟度的路线。

(4) SOA 监管 设计组织和流程,确保 SOA 的设计原则在 IT 生命周期中得以贯彻,管理服务生命周期中各种迁移的合理性等。

面向服务的开发方法并不是全新的方法学,它是现有方法学的继承和发展。一方面,原有的方法学并不能解决服务概念引入带来的问题,如如何识别服务、如何定义服务;另一方面,服务是一个水平而不是垂直概念。在服务分析和设计的过程中,需要处理服务和现有方法学的关系,如业务流程管理(business process management,BPM)和服务、企业架构和 SOA、服务和对象等。因此,服务的分析和设计最主要的职责在于识别服务、定义服务和实现服务,并指导如何和其他方法学相结合。

对于面向服务的开发方法,一方面是面向服务的分析和设计通过与 BPM 结合将业务分解为各种类型的服务,可以作为业务蓝图指导企业架构的设计;另一方面,企业架构设计的结果,又是服务实现的重要依据。各种 BPM 构成要素是面向服务的分析和设计的重要输入,如业务组件、业务流程和业务目标是服务识别的重要依据,而业务指标、业务数据、业务规则等是服务分析和实现的重要依据。

对于面向服务的开发方法,在原理层面很多面向对象开发的设计原则,如抽象、关注隔离等被 SOA 继承和发扬,并应用于服务的定义和实现中;而在操作层面上,服务模型为面向对象的开发进行类和对象设计提供了业务蓝图和企业架构蓝图。

参考文献

1. 冯玉琳,黄涛,金蓓弘. 网络分布式计算与软件工程. 北京:科学出版社,2011

2. Papazoglou M P, Traverso P, Dustdar S, et al. Service-oriented computing: state of the art and research challenges. IEEE Computer, 2007. 40(11)

3. Bloomberg J. The seven principles of service-oriented development. XML and Web Services Magazine, 2002

4. 毛新生. SOA 原理 · 方法 · 实践. 北京:电子工业出版社,2007

(魏峻)

mianxiang fuwu de tixi jiegou

面向服务的体系结构(service oriented architecture,SOA) 一种软件架构模型。它可以根据需求通过网络对松散耦合的服务组件(粗粒度应用构件)进行分布式部署、组合和使用。SOA 将应用程序的不同功能单元(称为服务)通过这些服务之间定义良好的接口和契约联系起来。接口是采用独立于实现服务的硬件平台、操作系统和编程语言的中立方式进行定义的。这使得构建在各种各样的系统中的服务可以以一种统一和通用的方式进行交互。国际标准化组织 OASIS 的标准定义为:SOA 是一种组织与利用分布式服务能力的范式。SOA 是将服务需求与服务能力结合在一起的机制,从而实现服务需求与服务能力的匹配,使服务能力满足不同服务需求。可见性、交互性和实效性是 SOA 的关键特性。SOA 系统的强调基于消息通信的松耦合、基于开放标准互操作、大粒度重用、基于服务的敏捷灵活开发、动态组合扩展、基本的自治等特征,是企业及其信息系统迅速适应市场需求变化、及时提供应用领域解决方案的有效技术手段和理想模式。

SOA 的概念于 1996 年由 Gartner 公司首次提出,由于当时技术水平和市场需求尚不具备真正实施 SOA 条件,SOA 并未引起广泛关注。SOA 发展历程,大致分为三个阶段。20 世纪 90 年代,可扩展置标语言(XML)出现。XML 规定了服务之间以及服务内部数据交换的格式和结构,为 SOA 的兴起奠定

了坚实基础。2000 年后,随着**互联网 Web 服务**及其标准和规范的发展,特别是简单对象访问协议 SOAP、Web 服务描述语言 WSDL、通用服务发现和集成协议 UDDI 的发展形成了基于互联网的开放通信框架。2005 年以后,互联网迅速出现了大量基于不同平台和语言开发的 Web 服务组件,迫切需要一种新的面向服务的分布式 Web 计算架构有效地管理服务的交互、复用和组合,于是,面向服务的体系结构 SOA 迅速发展起来。在全球范围内,各大 IT 公司开始相互协作,制定了中立的 SOA 标准,主要体现为三个重量级规范:服务组件架构 SCA、服务数据对象 SDO、Web 服务策略 WS-Policy。SCA 与 SDO 构成了 SOA 编程模型的基础;WS-Policy 建立了 SOA 组件之间安全交互的规范。这三个规范的发布,标志着 SOA 进入了实施阶段。目前,SOA 已成为一种主流的软件架构。

面向服务的体系结构 SOA 的研究技术内容主要包括以下几个方面:

(1) SOA 体系结构　SOA 逻辑上是个多层参考体系结构,可划分为五个横向层和四个纵向层。五个横向层自上至下分别为:服务使用层、业务过程层、服务层、服务构件层、资源层。跨越所有横向层的四个纵向层包括:集成层、服务质量(QoS)层、信息结构层、管理与政策层。其中,资源层包含运行在 IT 环境中的所有应用系统和服务构件,它们是构建 SOA 的资源;服务层由 SOA 所有服务组成,既有原子服务,也有组合服务,每个服务都提供规范描述,服务使用者依据规范描述引用服务功能;业务过程层管理业务过程生命周期,服务业务过程由业务需求驱动,通过组装服务层的服务实现企业或组织的业务逻辑,利用数据流与控制流技术组合与编排服务或业务过程;业务过程层桥接业务需求和面向服务的解决方案;服务使用层也称表现层,作为前端系统支持快速构建和访问 SOA 的业务过程,SOA 用户或程序通过服务使用层与 SOA 系统进行交互。纵向层从不同侧面对横向层提供支撑。其中,集成层提供集成功能,支持相邻层间通信、服务调用和 QoS 实施,能够路由、传送服务请求和服务响应。企业服务总线 ESB(enterprise service bus)可对这些集成功能提供有力支持。QoS 层负责监控、管理 SOA 的非功能性 QoS 需求,如可靠性、可用性、可管理性、可扩展性和安全等。信息结构层主要涵盖企业和行业的数据标准规范和元数据,包括数据发现、数据分析和**数据挖掘**等各种模型架构。管理与政策层涵盖了业务操作生命周期所有管理方面的内容,包括能力、性能、安全与监控等功能,支持服务层协议(service-level agreement,SLA)管理。

(2) SOA 服务组件架构技术　服务组件架构(service component architecture,SCA)通过服务组件,使用统一标准接口对不同类型组件进行封装和调用,可以屏蔽服务的具体实现,并对外提供服务接口。SCA 使用户通过服务组件的方式来构建企业应用,不再直接面对具体实现技术细节。

(3) 服务数据对象技术　服务数据对象(service data object,SDO)提供统一的数据模型整合服务和应用系统数据编程模型,支持 SOA 中服务间数据交换。在 SOA 中,SDO 是数据编程结构和应用程序设计接口(API),能够提供统一方式来访问处理异构数据源的数据,以此实现简化数据编程的目的,使开发人员集中于业务逻辑问题,而非访问数据的具体技术细节。

(4) 服务组合　SOA 服务组合技术主要分为服务编排(orchestration)和编舞(choreography)技术,从两个方面解决服务组合以及流程自动化等问题,以此改善业务系统随需应变的敏捷性。服务编排以局部单方视觉描述可执行业务过程内部与外部各服务之间交互。WS-BPEL(Web Services Business Process Execution Language)是 OASIS 专为整合 Web 服务而制定的规范标准,定义了业务流程以及如何与合作伙伴服务进行交互。服务编舞以全局多方视觉描述服务参加者间交互协作协议。WS-CDL(Web Service Choreography Description Language)是 W3C 提出的基于 Web services 相应技术规范,是描述两个对等参与者如何进行协作的 XML 描述语言,通过定义全局行为和消息交换顺序来实现参与者的共同目标。

(5) SOA 生命周期管理与开发方法　采用面向服务的思想和原则,涉及各生命周期和相关层面开发技术规范,主要包含面向服务的需求分析、服务建模以及服务开发实现方法等。SOMA(service-oriented modeling and architecture)是由 IBM 提出的、支持 SOA 开发的主要技术。SOMA 用于指导如何识别、规范和实现 SOA 的服务、服务流、服务组合以及保证 SOA 系统 QoS 的相关构件。

(6) SOA 实现技术　Web 服务技术是 SOA 的主流实现技术。Web 服务基本协议栈是 SOA 实现基础,包括:传输协议、消息协议、服务描述协议和服务发布和发现协议等。主要相关技术标准包括:SOAP 协议、HTTP、HTTPS、SMTP、XMPP、WSDL、UD-

DI 等。其中,SOAP 及 WSDL 由国际标准组织 W3C 负责,UDDI 由国际标准组织 OASIS 负责。

(7) 企业服务总线 ESB　是实现 SOA 分层目标所必需的基础功能部件的集成平台,结合了中间件技术与 XML、Web 服务等技术。ESB 主要包括服务元数据管理、服务通信与传输、中介协调机制、多种服务集成、服务和事件管理、服务交互、服务安全、服务质量、服务性能与可用性等级管理等功能。ESB 提供一种开放的、基于标准的消息机制,实现企业应用不同消息和信息的准确高效安全传递,完成粗粒度应用服务和其他组件之间的互操作,满足大型异构企业环境的集成需求。

面向服务的体系结构 SOA 对计算机技术和 IT 产业产生了深刻变革,包括 Web 2.0、软件即服务 SaaS、云计算、IT 系统体系结构、系统开发方法、软件运行模式等。由于 SOA 能够提供快速、敏捷的集成解决方案,SOA 的市场需求巨大,已经广泛应用于各行业的不同领域,如金融、医疗卫生、保险、电信、电子政务、电子商务、生产制造、生物信息、云计算以及复杂分布式计算系统等。特别是,SOA 将对现代服务业发展产生极大的推动促进作用。

经过近几年的发展与积累,SOA 基础理论方法和技术体系已经初步形成,SOA 技术在不断向广度与深度发展,其主要研究和发展趋势有:

(1) SOA 应用精细化　随着 SOA 应用的不断普及,各类服务内容和业务将根据应用领域越分越细,形成越来越多的垂直应用细粒度服务,满足各应用领域的客户更细微的需求。

(2) SOA 与**云计算**相互融合　云计算是 SOA 解决方案的交付支撑技术,提供 SOA 系统的部署和运营环境,云环境的虚拟化技术将对 SOA 系统的 QoS 提供保障;同时 SOA 技术与云计算结合,将促进各种云服务(XaaS)的开发和运营。

(3) SOA 系统自治化　SOA 技术将于自治计算和云计算技术相结合,针对顾客和应用需求以及运行环境的不断变化,SOA 可进行动态演化,快速地进行资源与服务重构,形成有效的解决方案。

(4) SOA 系统泛在化　随着移动计算、云计算、**物联网**、务联网的不断发展,未来互联网环境下的 SOA 系统将与移动互联技术相结合,使人们能够随时随地随需通过各类移动设备,便捷高效地使用服务。

(5) SOA 技术标准渐趋成熟　目前 SOA 相关技术标准繁多,已经引发相互之间工作的不协调。国际上的标准组织正在把 SOA 架构和模型中最核心的东西统一起来,使相关技术标准制定组织互相协调,互通有无,促进 SOA 技术标准走向融合与成熟。

参考文献

1. Erl T. Service-oriented architecture: concepts, technology, and design. Prentice Hall PTR, 2005

2. Rosen M, Lublinsky B, Smith K T, et al. Applied SOA: service-oriented architecture and design strategies. Wiley Publishing, Inc., 2008

(臧天仪　徐晓飞)

mianxiang shuju jiegou fangfa

面向数据结构方法(data structure-oriented method)　一类侧重从数据结构方面去分析和表达软件需求,进行软件设计的开发方法。该方法从数据结构入手,分析信息结构,并用数据结构图(特指该类方法所用的图形描述工具,例如 Jackson 结构图,Warnier 图)来表示,再在此基础上进行需求分析,进而导出软件的结构。

应用领域的信息域一般包括信息流、信息内容和信息结构等部分。在软件需求分析过程中,根据对信息域分析的侧重点不同就形成了不同的开发方法。面向数据流的开发方法以分析信息流为主,用数据流图来表示信息流;而面向数据结构的开发方法则以分析信息结构为主,用数据结构图来表示信息结构,并以此为基础进行需求分析,进而导出软件的结构。

面向数据结构的开发方法包括分析和设计两个过程,数据结构在整个过程中起着重要作用。由于很多应用领域的信息都有层次分明的信息结构,系统的输入数据、内部储存数据及输出数据都存在着层次性,并且是相对独立和可区分的,因此,在需求分析过程中可以利用数据结构来分析和表示问题的信息域。在软件设计过程中,不同性质的数据结构往往可以用具有相应的控制结构的程序进行处理。重复性的数据总是用具有循环控制结构的软件来处理,而选择性的数据则由具有条件处理部分的软件来处理,业已证明,仅用三种结构成分即顺序、选择和循环就可以表示所有具有单出口与单入口的程序,因此,可以将具有层次性的数据结构映射到结构化的程序上。

20 世纪 70 年代初,在关于数据结构基础的论述中已出现了面向数据结构的设计思想。

1974 年 J. D. Warnier 发表了《程序的逻辑构造》一文，提出了一种表示数据层次结构的图形工具，即 Warnier 图。他利用顺序、选择、重复三种结构成分来表示数据的层次结构，进而导出程序的结构。1975 年 Michael Jackson 在《程序设计的原则》一文中将输入数据与输出数据的结构在相应层次上对应，并系统地提出了将数据结构映射到程序结构的实用技术。从此，面向数据结构的开发方法便发展起来。

早期的面向数据结构的开发方法主要是针对程序的过程设计而言的，80 年代初逐步发展为较完善的系统化软件开发方法。Warnier 将程序的逻辑构造(LCP)法进一步完善，提出了系统的逻辑构造(LCS)法，用逻辑和形式化方法严格地表示了数据结构，这样便有利于自动生成伪代码，进行系统的校验及优化。同时，Jackson 也将结构化程序设计(JSP)扩充为 Jackson 系统开发(JSD)方法。

1981 年和 1983 年 Ken Orr 对 Warnier 方法进行了扩充，分别发表了《结构需求定义》与《数据结构化程序设计》两文，形成了一种易于理解，并且文档比较完善的系统化开发方法，这就是数据结构化系统开发(DSSD)方法。这一方法涉及了信息域的所有属性：信息流、信息内容和信息结构。这种方法首先研究应用环境，引入了数据流中的分析方法，以此强化对系统功能特性的分析。同时，也采用了公式化和综合性的设计方法。

虽然各种面向数据结构的开发方法都有自己独特的表示方式及开发过程，但他们都具有一些共同特点：①这类方法开始于需求分析，并将分析结果作为设计基础，但各种方法并不明显地区分软件的结构设计与过程设计，它们都较早地进入软件的过程表示；②各种方法都为分析人员提供手段来标识关键信息对象(也称为实体或项)和操作(也称为动作或过程)；③各种方法都假定信息结构具有层次性；④基本上都采用顺序、选择和重复构造成分表示数据结构；⑤各种方法都提供了一组将层次化的数据结构映射到程序结构的步骤。

面向数据结构的开发方法适用于具有良好定义的，层次分明的信息结构的领域。典型的例子有：商业信息系统开发，其输入和输出有明显的数据结构，并且共同使用同一层次的数据库；操作系统开发，其数据结构由许多有确定结构的表格、文件等构成；CAD/CAM 系统的开发，这方面需要复杂的数据结构用于信息的储存、变换和处理。此外，在工程领域、计算机辅助教育、组合问题求解及其他许多领域都可使用面向数据结构的开发方法。

参考文献

Pressman R S. Software engineering：a practitioner's approach. 3rd ed. New York：McGraw-Hill Inc.，1992

（郝克刚　葛玮）

mianxiang wenti yuyan

面向问题语言(problem-oriented language)

为了易于描述和求解某类指定问题而专门设计的一种**非过程语言**。利用它在计算机上解题时，只要指出问题、输入数据和输出格式，就可由相应的计算机系统给出结果。解题过程如图 1 所示。

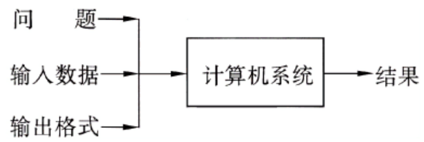


图 1　面向问题语言的解题过程

面向问题语言的同义语有面向应用语言，专用语言，专业化应用语言，或者专业化应用领域的语言。与面向过程语言相比，面向问题语言不要求程序人员描述问题的计算过程或算法。其主要特点是：①申述性。程序人员只要指明“做什么”(及有关参数)，而不必详细说明“如何做”，后者由系统自动解决。②对用户友善。面向问题语言一般是为最终用户设计的，专业性强，用户稍经训练便会使用。③效率高。

早在 20 世纪 60 年代，数值计算的面向问题语言(如线性代数语言)和非数值计算的面向问题语言(如机床控制语言)曾经蓬勃发展。自 60 年代末以来，随着数据库、软件工程、微型计算机和网络等迅猛发展，出现了很多功能强大的面向问题语言，如数据库检索语言，以至诞生了含义更为丰富的名字——**第四代语言** 4GL。

面向问题语言的主要缺点是：①无标准、难移植；②通用性差；③适应性差，不适应计算机资源的扩充升级。

参考文献

1. Truitt T D, Mindlin S B. An introduction to nonprocedural languages：Using NPL. McGraw-Hill Osborne Media，1983

2. 郭浩志. 程序设计语言概论. 长沙：国防科学技术大学出版社，1989

（郭浩志）

mianxiang zhinengti chengxu sheji

面向智能体程序设计(agent-oriented programming,AOP) 以智能体作为基本编程和运行单元的一种程序设计风范(参见**多智能体系统**)。在面向智能体的程序设计中,构成软件系统的基本运行单元是一个个的软件智能体,每个智能体封装了知识、信念、承诺、能力、意图等认知部件,实现了一个或者一组相对独立的功能。这些认知部件的取值定义了智能体的内部状态。智能体根据其内部状态选择适当的行为来执行。智能体的内部状态如何变化、智能体如何根据其内部状态实施自主的行为是由智能体自身来决定的。不同智能体之间通过高层的社会性交互实现相互作用。一般地,这种社会性交互是通过智能体之间的言语行为,借助于**智能体通信语言**来实现的。智能体间的言语行为实际上是一种特殊的消息传递格式,它不仅能清晰地表示智能体之间交互的意图,而且能详尽地表述交互的内容,从而能有效地促进智能体之间复杂的协作。因此,对智能体程序设计的本质是要在设计阶段由程序设计人员描述和定义构成智能体各个认知部件的具体内容以及智能体之间的社会性交互。

面向智能体程序设计思想最早由美国斯坦福大学的学者 Shoham 在 20 世纪 90 年代初期提出,并设计了一个基本的语言设施 Agent-0 及其解释器。至今,人们已经提出了许多面向智能体的程序设计语言,并采用编译或解释的方式来支持程序的运行。根据这些语言实现技术的差异,可以将现有的面向智能体程序设计语言大致分为以下 3 类:①基于 Lisp 技术 这类语言采用 LISP 的语法形式,用类似于 LISP 的语句来表示智能体的内部组成成分、行为以及智能体之间的交互。它们具有良好的理论基础和严格的语义定义并采用符号处理的方式来解释和执行所编写的程序,代表性工作包括:由 Shoham 提出的 AGENT-0,由 Thomas 提出的 PLACA 等;②基于逻辑技术 这类语言采用各种形式的逻辑系统来表示和定义智能体的结构和行为。它们具有良好的理论基础和严格的语义定义,将智能体视为是一个定理证明器,将智能体的执行过程视为是一个定理证明过程。代表性工作包括:由 Michael Fisher 提出的并发 METATEM,由 De Giacomo 和 Yves Lesperance 等人提出的 CONGOLOG 等;③基于对象技术 这类语言通过对现有**面向对象语言**进行扩充,为面向智能体的程序设计提供语言支持。它们通常将智能体视为是一个特殊的对象,通过提供一组封装了智能体内部功能的预定义软部件来支持对智能体的编程和实现。一般地,它们将用户编写的程序直接编译为可执行的代码在目标平台上运行,或者通过预编译生成由面向对象程序设计语言所描述的程序代码,然后再由该面向对象程序设计语言的编译器对它进行编译以生成最终可执行的目标代码。代表性工作包括:由 Agent-Software 公司提供的 JAL(Java Agent Language)等。

参考文献

1. Shoham Y, Agent-oriented programming. Artificial Intelligence, 1993, 60(1): 51-92

2. 毛新军. 面向主体的软件开发. 北京: 清华大学出版社,2005 (毛新军)

miaoshu luoji

描述逻辑(description logic) 用术语定义概念,并用概念描述性质的**知识表示**方法的逻辑系统。知识表示一种有效方法为:先定义应用领域("世界")的有关概念(术语),然后用这些概念表达该领域中对象和个体的性质(世界描述)。描述逻辑为这种知识表达方法提供理论基础(包括形式语义和推理机制)。

描述逻辑有一个描述语言来定义应用领域的术语。术语包括概念(个体集合)和角色(个体间的二元关系)。初等的描述是原子概念(A,B)和原子角色(R),复杂的描述(C,D)由初等描述用概念构造算子归纳地产生。不同的描述逻辑提供不同的概念构造算子集。通常认为具有实际意义的最小描述语言 AL 的概念描述为:$C,D \to A \mid T \mid \bot \mid \neg A \mid C \sqcap D \mid \forall R.C \mid \exists R.T$。给定一个解释 I(包括非空论域 Δ^I,并给原子概念 A 分配 $A^I \subseteq \Delta^I$,给原子角色 R 分配 $R^I \subseteq \Delta^I \times \Delta^I$),下面的公式给出 I 在 AL 的概念描述上的扩充(亦即给出了它们的形式语义):

$T^I = \Delta^I$ (顶概念)

$\bot^I = \emptyset$ (底概念)

$(\neg A)^I = \Delta^I \backslash A^I$ (原子概念的非)

$(C \sqcap D)^I = C^I \cap D^I$ (概念的交)

$(\forall R.C)^I = \{a \in \Delta^I \mid \forall_b (a,b) \in R^I \to b \in C^I\}$ (值限制)

$(\exists R.T)^I = \{a \in \Delta^I \mid \exists_b (a,b) \in R^I\}$ (有限存在量化)

描述逻辑中的概念可以翻译为谓词逻辑的合式公式,所以描述逻辑是谓词逻辑的子集。但是描述逻辑的表达更简洁,且易于开发有效的推理算法。

基于描述逻辑的知识表示系统简称 *DL* 系统。DL 系统的**知识库**包括 TBox 和 ABox。描述逻辑可以为复杂概念"定义"它的符号名,一组这样的定义(型如 $C \equiv D$)构成描述逻辑的术语(又称 TBox)。若术语中的定义含有循环,我们需要使用定点语义。何种术语具有定点模型是一个重要的理论问题。描述逻辑还提供世界描述(又称 ABox),即一组关于个体所属概念或所参与角色的断言(如 $C(a)$,$R(b,c)$ 等)。虽然 ABox 看起来像仅含 1-2 元关系的关系数据库,但它采用"开放世界语义",这意味着描述逻辑的推理要比关系数据库的查询复杂。

DL 系统能够从知识库存放的显式知识推出隐含的知识。*DL* 系统推理是逻辑推导,推理任务包括对概念的推理(如概念的可满足性,概念间的包容性等)和对 TBox 和 ABox 的推理。(只要描述语言容许并和非)所有这些推理可归结为一个主推理问题:ABox 的一致性检查。

有些 *DL* 系统还使用规则(如 $C \Rightarrow D$,意为:"若个体 a 属于概念 C,则 a 也属于概念 D")来表达知识。规则的过程性语义比较直观;而为了给出说明性语义,需要扩充描述语言。

描述逻辑的一个研究重点在于其表达能力和计算复杂性之间的权衡。概念构造算子越丰富,表达能力就越强,但推理计算也就越复杂,甚至使核心的推理问题(概念之间的包容性)成为不可计算的。即使对表达能力很小的语言,该问题也可能是难计算的。为了实用,*DL* 系统的实现通常使用复杂的优化技术。描述逻辑最重要的应用领域是语义 Web。语义 Web 体系架构中的核心层——本体层是以描述逻辑为逻辑基础的。

参考文献

Baader F, et al, eds. The description logic handbook. Cambridge University Press, 2003 (刘椿年)

minzu yuyan wenzi xinxi chuli

民族语言文字信息处理(national language information processing) 以中国少数民族语言文字为主要研究对象,实现用少数民族语言文字在人与计算机之间进行有效通信的各种理论、方法和技术。

目前,中国 55 个少数民族使用着上百种语言,这些语言分属 5 个语系、10 个语族、16 个语支。有 24 个民族有与自己的语言相一致的文字,共 33 种文字。

20 世纪 80 年代以来,开启了我国少数民族语言文字信息化建设的进程。蒙古语、藏语、维吾尔语、哈萨克语、柯尔克孜语、朝鲜语、彝语、傣语等走在我国民族语言文字信息化的前列,取得了许多优秀的科研成果。

民族语言文字信息处理可以分为**民族语言信息处理**和**民族文字信息处理**两大类。

民族语言信息处理 是以少数民族的言语和文本信息为研究对象,进行现代民族语言的语法体系、词汇计量、语料库语言学等基础研究,以及民族语言文字的编辑排版、文本校对、信息检索、**机器翻译**、**信息提取**、**自动文摘**、辅助教学、远程教育和电子词典等应用研究。民族语言信息处理又可分解为民族语言言语信息处理和民族语言文本信息处理。

民族语言言语信息处理 是指利用计算机科学技术处理民族语言的言语信息,内容主要包括言语识别、言语合成、文语转换、言语标注等,技术原理与汉语言语信息处理相同。其中民族语言言语标注是指建立民族语言的语音标注语料库,主要包括文字标注和音节标注两部分。文字标注是将语音信息用文字记录下来,以便提供给识别系统使用。音节标注主要为声调标注,是针对各民族语言的不同发声方法,进行声调标注。

民族语言文本信息处理 旨在帮助人们在大量文献资料中迅速获得所需要的信息。目前的文本信息处理系统大多采用统计语言模型和语言浅层分析规则的策略,这样就避开了自然语言理解难以解决的问题,把一些相对成熟的语言分析技术应用到语言工程中。目前我国少数民族语言文本信息处理技术主要包括民族语言文本的编辑排版、信息检索、信息抽取、机器翻译、文本分类、文本校对、自动文摘,以及信息过滤等方面内容。

(1) 民族语言文本编辑排版 不同的民族语言文字往往具有不同的编辑排版规则和特征。例如,藏文的文本编辑基准线是在文字上部(不同于汉语和英语的基准线位于文字下部)、蒙古文采用文字纵向排版格式等。民族语言文本编辑排版是指实现民族文字、汉字、英文等多文种文字的输入编辑、排版、校对、组版等相关技术内容。

(2) 民族语言机器翻译 中文信息处理中较为复杂的一项关键技术。由于受到语言自身以及使用人群等多种因素影响,使得翻译系统的研制既涉及语言学研究,也与其他领域的知识有关。总体来说,主要分为基于规则的和基于语料库统计的两种处理

策略。

(3) 民族语言文本校对　针对一些基于拼写式的少数民族语言文字,在实际书写过程中难免会出现错拼、漏拼及不符合正字正音规则的现象,借助自然语言处理技术,实时检查少数民族语言文本字词错误,对文本的字词、语法、语义错误进行综合校对,构建易混淆词典对一些字词错误提供实时动态的纠错建议。

民族文字信息处理　是指通过对少数民族文字的字符属性、字符编码、字型字体,以及键盘输入的描述与定义,使之能够被计算机系统所识别,从而实现民族文字的数字化输入与存储。

(1) 民族文字字符属性　对民族文字字符本身的元数据描述,包括字符名称、所属类别、读音、形状,以及字际关系、文法、编码等多种信息。从信息处理的角度来看,民族文字根据其构成形式大致可分为图画文字、象形文字、音节文字和拼音文字。

(2) 民族文字字符编码　我国各少数民族文字字符和符号编码的总称,包括民族文字字符、标点符号、图形符号、数字等。它主要用于少数民族文字字符和符号的计算机存储、传输等。常见的字符编码集有 ASCII 字符集、Unicode 字符集等。目前 Unicode ISO/IEC 10646-1:2000 中已收入中国少数民族编码字符集包括蒙古(传统回鹘蒙古文、托忒蒙古文、锡伯文和满文四种文字)、藏、维吾尔、哈萨克、柯尔克孜、彝文编码字符。八思巴文字符集国际标准已完成提案,西双版纳傣文编码字符集国际标准已经完成并提交 ISO/IEC JTC1/WG2 表决,藏文以垂直组合字符为编码单元已在国际标准框架下实现了藏文字符扩充集、点阵字型和键盘标准;新疆维吾尔自治区政府启动《维哈柯标准化》项目,定义了 ISO/IEC 10646.1:2000 中的阿拉伯文字符与维吾尔、哈萨克、柯尔克孜文字符的对应关系并向 ISO/IEC JTC1/SC2/WG2 提出对 ISO/IEC 10646 的补充提案。

在 2003 年制定的 Unicode 4.0 国际标准中我国少数民族文字的编码字符包括:①藏文基本集 Tibetan (0F00-0FFF),②蒙古文基本集 Mongolian (1800-18AF),③傣文 Tai Le(1950-197F),④西双版纳傣文 Tai Lue (1980-19DF),⑤彝文基本集 Yi Syllables&Yi Radicals (A000-A48F, A490-A4C6),⑥八思巴文 Phags-pa(A840-A87F),⑦朝鲜文 Hangul(AC00-D7AF)等多个少数民族文字编码字符集。

我国现行主要少数民族文字编码字符集标准包括:①藏文编码字符集标准:《GB/T 20542—2006 信息技术藏文编码字符集扩充集 A》、《GB/T xxxxx—200x 信息技术藏文编码字符集扩充集 B》已获批。②维吾尔文编码字符集标准:《信息技术维吾尔文、哈萨克文、柯尔克孜文编码字符集》已获批。③蒙古文编码字符集标准:《蒙古文编码国际标准字符集及控制符使用细则》已获批。④彝文编码字符集标准:完全采用 ISO/IEC 10646:2003 的彝文编码。包括:彝文字母(Yi Syllables),编码空间:U+A000~U+A48C;彝文部首(YiRadicals),编码空间:U+A490~U+A4C6。⑤傣文编码字符集:完全采用 ISO/IEC 10646:2003 的傣文编码。包括:德宏傣文((Tai Le),编码空间:U+1950~U+197F;西双版纳傣文(New Tai Lue),编码空间:U+1980~U+19DF。

(3) 民族文字字型字体　在信息处理过程中由计算机所显示或打印出来的少数民族文字字符的形体、类型等相关信息。常用字体字型设计规范有国家语委立项的《民族文常用字体字型设计与规范》,包括两款藏文、维哈柯文、蒙古文、彝文常用字型规范,一款为正文字,一款为标题字。我国目前少数民族文字的主要点阵字型标准包括:

①藏文点阵字型标准:《信息技术藏文编码字符集(基本集及扩充集 A) 24×48 点阵字型》,包括吾坚琼体、朱匝体、白祖体;《信息技术藏文编码字符集(扩充集 B) 24×48 点阵字型》吾坚琼体。

②维哈柯文点阵字型标准:《信息交换用维吾尔文、哈萨克文、柯尔克孜文 32×32 点阵字型》,包括塔里克白体、塔里克黑体、正文白体、正文黑体、如克白体、如克黑体、库非自体、库非黑体、苏鲁斯黑体、苏鲁斯白体、地万黑体、地万白体。

③蒙古文点阵字型标准:蒙古文 16×16 点阵字型白体、新五号;蒙古文 32×32 点阵字型白体、新五号;锡伯文 32×32 点阵字型白体、小黑体、大黑体;满文 32×32 点阵字型白体、小黑体、大黑体。

④傣文点阵字型标准:德宏傣文 32×32 点阵字型克炳珍白体、克炳珍黑体;西双版纳新傣文32×32 点阵字型岩温胆白体、岩温胆黑体、岩化白体、岩化黑体。

(4) 少数民族文字输入法　计算机输入少数民族文字字符,并以民族文字字符编码文件流形式存储及处理少数民族文字字符信息的软件或系统。通常采用键盘输入和文字识别(参见**文字识别 OCR**)两种技术。国家已颁布的民族文字键盘布局标准包

括：①藏文：GB/T 17543—1998 信息技术 藏文编码字符集(基本集)键盘字母数字区的布局；②藏文：GB/T 22034—2008 信息技术 藏文编码字符集键盘字母数字区的布局；③维吾尔文：GB/T 12510—1990 信息处理交换用维吾尔文字符集键盘的字母区布局；④蒙古文：GB/T 8046—1987 信息处理交换用蒙古文字符集键盘的字母区布局；⑤傣文：德宏傣文、西双版纳新傣文键盘布局标准，已通过审定，尚未公布。

少数民族语言文字的信息化是国家语言文字信息化的重要组成部分。促进与规范少数民族语言文字信息处理进程，建设与共享少数民族语言文字信息资源，推进少数民族语言文字信息处理研究成果的技术转化及应用，使其更好地服务于少数民族地区的文化教育与经济建设，是我国信息化建设的长久任务之一。

参考文献

1. 戴庆厦，赵小兵. 中国少数民族语言文字信息处理研究与发展. 北京：民族出版社，2010

2. 戴庆厦. 中国少数民族语言文字. 北京：民族出版社，2006

3. 中国科学院软件研究所. 少数民族语言文字信息化. 第三届中文数字化论坛，2005

（赵小兵　翁彧）

minjie ruanjian kaifa

敏捷软件开发(agile software development)

一种基于面向对象技术的软件开发风范。以提高面临迅速变化的市场需求而快速开发软件的能力。

敏捷软件开发的基本思想是“满足工程设计标准的唯一文档是源代码清单”。因此，软件项目的几乎所有步骤都是设计的一部分，包括高层设计、模块设计、体系结构设计、软件编码、测试/调式等；并且，与传统制造业相比，软件开发中的“制造”是相当廉价的，只须编译器和连接器等，不需要相对独立的制造团队。

基于以上思想，从快速响应市场需求的角度，敏捷软件开发提出了12条开发原则，用于指导敏捷实践；11条设计原则以及相关的设计模式，用于指导敏捷设计。

12条开发原则是：①最优先要做的事情是，尽早地、持续地交付有价值的软件，以使客户满意；②欢迎客户依据市场需求提出有关产品需求的变更，即使到了开发后期，以便利用变化为客户创造竞争优势；③频繁交付可工作软件，其时间间隔越短越好；④在整个项目开发期间，业务人员和开发人员必须天天在一起工作；⑤围绕激励人员，包括给他们提供所需要的环境和支持，开展项目的有关工作，并信任他们能够完成所承担的工作；⑥交流，特别是面对面的交流是在开发团队内部最有效果的、最有效率的传递信息的方法；⑦把可工作的软件作为度量进度的首要标准；⑧责任人、开发者和用户能保持一种长期的、恒定的可持续开发速度；⑨不断关注优秀的技能和设计，增强敏捷能力；⑩使未完成工作最小化，简单是根本的；⑪好的体系结构、需求和设计，出自自己组织的团队；⑫每隔一段时间，团队应对如何才能有效的工作进行反思，然后对自己的行为进行适当的调整。

为了支持实施以上12条开发原则，敏捷软件开发给出了11条设计原则。其中5条是有关基本构造块的设计原则，6条是有关打包的设计原则。

基本构造块的设计原则为：①单一职责原则，就一个类而言，应该仅有一个引起它变化的理由，即一个类应是功能内聚的；②开放-封闭原则，软件实体(类、模块、函数等)应该是可扩展的，但是不可修改的；③Liskov替换原则，子类型必须能够替换它们的基类型；④依赖倒置原则，高层模块不应该依赖低层模块，二者都应该依赖抽象；抽象不应该依赖细节，细节应该依赖抽象；⑤接口隔离原则，不应迫使客户端依赖它们不用的方法。

打包的设计原则为：①复用发布等价原则，复用的粒度就是发布的粒度；②共同复用原则，一个包中的所有类应该是共同复用的，如果复用了包中的一个类，那么就要复用包中的所有类；③共同封闭原则，包中的所有类对于同一类性质的变化应该是共同封闭的；一个变化若对一个包产生影响，则将对该包中的所有类产生影响，而对其他包不产生任何影响；④无环依赖原则，在包的依赖关系图中，不允许存在环；⑤依赖稳定原则，朝着稳定的目标进行依赖；⑥稳定抽象原则，包的抽象程度应该和其稳定程度一致。

以上提及的11条设计原则，都是**软件工程**一些原则的特例。在应用中，不能无条件地遵循这些原则，以免发生应用上的错误；也不能“过分地”遵循这些原则，以免导致不必要的复杂性。

为了支持实现以上设计原则，敏捷软件开发还给出了一些相关的设计模式。

可见，如果能很好地运用敏捷软件开发这一开

发风范，就可逐步提高并具备一种应对迅速变化的市场需求而快速开发软件的能力。 （王立福）

minglingshi yuyan

命令式语言（**imperative language**） 通过指明一列可执行的运算及运算的次序来描述计算的语言。又称强制式语言或**过程式语言**。

命令式语言以命令驱动的冯·诺依曼式体系结构的计算机为背景。**机器语言**与**汇编语言**是最早问世的命令式语言。FORTRAN、ALGOL、COBOL、PASCAL、C、Ada、C ++ 、C#、Java、Python 等高级语言也都是命令式语言，其变量对应于存储单元，对变量的访问就是对相应存储单元的访问。各个语句在程序中的顺序以及转向语句等控制语句则明确规定了机器的执行步骤。

高级语言中每种成分几乎都要翻译成若干低级机器指令。由于存储空间有限，几乎所有程序都要用重复语句重复执行某些语句，语句的重复执行使命令式语言程序的层次性受到很大的影响。

程序的执行反映组成计算的各计算步的依次执行。命令式语言程序的本质是重复地、按步地计算低级（非抽象）值并将之赋给变量（对象），这就迫使程序人员去关心比较低级的细节，而这不适用于设计复杂算法。因此，几十年来命令式语言一直向着隐蔽低级机器属性、提高程序层次与抽象性的方向发展。例如，面向对象语言就是把类（其实例为对象）作为程序的基本单元，将程序和数据封装其中，通过封装、继承、多态等手段提高软件的重用性、灵活性和扩展性。Ada、C ++ 、Java 等语言中提供的程序包、对象、类、重载、继承、并发处理（任务、线程）、异常、模板（类属）、泛型等机制不仅提高了程序设计的抽象性，也提高了程序的重用性与灵活性。

命令式语言无论如何发展，都是以冯·诺依曼式体系结构的计算机为其设计背景。例如，几乎所有命令式语言都要涉及全程变量、传地址参数（或变量参数、引用参数等），它们一方面可以提高程序的执行效率，另一方面也降低了程序的层次性和抽象性，使程序难以编写、阅读、分析与修改，也使得程序的正确性证明难以进行。随着在非冯·诺依曼式体系结构、基础理论、元器件、人工智能与**软件工程**等方面的不断发展，已有一些非命令式语言（如**申述式语言**）问世，函数式语言与逻辑语言就是其中的代表。

参考文献

1. Watt D A. Programming language concepts and paradigms. London: Prentice Hall, 1990

2. Watt D A. Programming language design concepts. John Wiley & Sons, 2004

3. Robert W. Sebesta. Concepts of Programming Languages. 9th ed. Addison Wesley, 2009

4. 徐宝文. 高级程序设计语言原理. 北京: 航空工业出版社, 1992

（徐宝文）

mingling yuyan

命令语言（**command language**） 交互计算机系统向用户提供的一种操作界面的语言。命令语言具有规定的词法、语法和语义，它以命令为基本单位来完成系统提供的各种独立工作任务。完整的命令集所构成的命令语言，反映了该系统向用户提供的功能。

一个命令语言驱动的系统，通常先向用户显示“命令提示符”；随后，用户可输入一条包括参数在内的命令，以实现某任务；在“命令结束符”输入后，系统执行该命令，并给出运行结果或报告出错情况；系统完成该命令后继续显示“命令提示符”，等待用户下一条命令。这种每次执行一条命令的交互式命令语言，与批处理方式的作业控制语言相比，具有明显的优点：简练，灵活，响应速度快，功能易扩充，便于用户根据前一命令结果选择以后的操作。

命令语言已广泛用于各类交互系统，诸如操作系统、正文编辑、数据库操纵、文献资料检索、电子邮件、飞机订票等。目前常用的 UNIX，DOS 操作系统均有命令语言操作界面，shell 作为 UNIX 的统一用户界面是一种典型的命令语言，其命令一般具有以下形式：

$ 命令名 可选项 文件名 其他参数

其中“ $ ”为系统的“命令提示符”；可选项是为增加功能而又不增多命令个数的扩展；文件名通常指该命令操纵的对象；命令行的结束符为“换行符”，未标出。shell 命令通常占一正文行，也可占多行（行尾使用“续行符”）；一行内可有多个命令，只需用“分隔符”分开。shell 还提供许多强的功能：后台作业、输入输出重新定向、shell 变量、命令替换、参数替换、管道线、元字符匹配及可用于编程的多种控制结构（条件、循环）等。

命令语言的设计应从应用的实际情况出发，主

要考虑功能需求及使用方便。从“人的因素”观点而论,设计时应考虑以下方面:命令结构一致性,命令名的可读性及缩写策略,提供命令组合、undo 命令、redo 命令、用户自定义命令及创建宏命令的能力。

命令语言方式的弱点是需良好的培训和记忆,有的命令语言过于复杂,有的出错处理功能较差。目前在广泛采用并不断改善命令语言交互方式的同时,还可用其他交互方式(如选单选项、表格填充、图形方式等),更加方便的自然语言界面正在研究中。

参考文献

1. Shneiderman B. Designing the user interface. Addison Wesley, 1987

2. 董士海. 计算机用户界面及其工具. 北京:科学出版社, 1994

(董士海)

mingti luoji

命题逻辑(propositional logic) 研究由命题与命题联结词构成的更复杂命题,以及这样构成的命题间的推理关系的逻辑。在命题逻辑中,不含命题联结词的命题称为原子命题。不再分析原子命题的内部结构,它们的性质只有真与假的区别。常用 1 和 0 分别代表真命题和假命题,并称集合{0,1}为真值集合。因此,可以将命题联结词看作真值集合上的运算。在命题逻辑中经常使用 5 个命题联结词:“非(¬)”,“与(∧)”,“或(∨)”,“如果…,则…(→)”和“当且仅当(↔)”。这 5 个命题联结词的运算表如表 1、表 2 所示。

表 1 命题联结词 ¬ 的运算表

p	$\neg p$
0	1
1	0

表 2 命题联结词 ∧, ∨, →, ↔ 的运算表

p	q	$p\wedge q$	$p\vee q$	$p\to q$	$p\leftrightarrow q$
0	0	0	0	1	1
0	1	0	1	1	0
1	0	0	1	0	0
1	1	1	1	1	1

由运算表可以看出,这 5 个命题联结词不是互相独立的。可用 ¬ 和→定义 ∧, ∨ 和↔。例如, $p\wedge q\equiv \neg(p\to\neg q)$; $p\vee q\equiv\neg p\to q$; $p\leftrightarrow q\equiv\neg((p\to q)\to\neg(q\to p))$。因此,可取 ¬ 和 → 为基本联结词。

常用英文字母 $p,q,r,\cdots$ 表示命题变元。公式的形成规则如下:

(1) 每个命题变元是公式;

(2) 若 A 是公式,则 $\neg A$ 是公式;

(3) 若 A,B 是公式,则 $(A\to B)$ 是公式;

(4) 每个公式都可以经有穷次应用(1),(2)和(3)获得。

只要为公式中的每个命题变元指定了真值,公式就有了确定的真值。如果对于公式 A 中的命题变元赋予任何真值, A 的值均为 1,就称 A 为重言式或永真式,记为 $\vDash A$。例如, $(p\to p)$ 就是一个重言式。

命题演算是命题逻辑的形式系统,它把重言式组成了一个完全形式化的公理系统。最早的命题演算是 G. Frege, G. Peano 和 B. Russell 于 19 世纪 70 年代至 20 世纪初建立起来的。

在命题演算中,取某些重言式为公理,并规定了某些推理规则,以便从公理出发推出所有的重言式。人们给出了许多不同的命题演算系统,虽然这些系统的公理和推理规则不同,但它们是等价的,即它们推出的定理集都是相同的。下面举出一个这样的命题演算系统。

取以下三种形式的公式为公理:

$A\to(B\to A)$

$(A\to B)\to((A\to(B\to C))\to(A\to C))$

$(\neg B\to\neg A)\to(A\to B)$

取分离规则为唯一的推理规则,即由前提 A 和 $A\to B$ 推出结论 B。命题演算的定理是这样定义的:

(1) 每个公理都是定理;

(2) 如果公式 A 和 $A\to B$ 是定理,则 B 是定理;

(3) 每个定理都可以通过有穷次应用(1)和(2)获得。若公式 A 是定理,则记为 $\vdash A$。

命题演算的公理都是重言式。推理规则保证,当前提被解释为真命题时,在同样的解释下结论也表示真命题。因此,命题演算的定理都是重言式。这个性质称为命题演算的可靠性,即形式推理可靠地反映了直观的逻辑推理。反之,每个重言式都是命题演算的定理。这个性质称为命题演算的完全性,即形式推理完全反映了直观的逻辑推理而无遗漏。这是 E. L. Post 于 1921 年首先证明的。

参考文献

1. 王宪钧. 数理逻辑引论. 北京:北京大学出版社,1982

2. Kleene S C. 元数学导论. 莫绍揆,译.北京:科学出版社,1984
（何自强）

moban

模板(template)　一种支持类属编程的机制,它允许人们使用参数化的类型来描述算法或类。人们在使用模板时,通过设定这些参数的取值就可以实例化得到函数或类。C ++ 中有两种模板:函数模板和类模板。

函数模板允许使用参数化的模板来描述一组函数,这些函数具有相似的结构,但是参数类型有所不同。函数模板可以把参数类型也参数化,并且通过对参数的实例化来得到普通的函数。比如,C + + 标准模板库中的函数 max(x, y) 回送 x 和 y 之间的较大者,但是它并没有规定这个函数的参数类型。它可以按照下面的方式定义

```
template <typename T>
T max(T x, T y)
{
    return x < y ? y:x;
}
```

这里定义的函数模板可以像普通函数一样被直接调用。**编译程序**将按照被调用参数的类型来确定类型参数的值。比如,如果我们使用整数值参数调用 max,比如 max(4, 5),那么编译程序根据参数的类型,确定实际上调用的是 max(int, int),因此,编译程序以 int 类型作为 T 的取值,得到上述模板的实例,等价于下面的函数

```
int max(int x, int y)
{
    return x < y ? y:x;
}
```

同样,我们可以使用其他类型的参数来调用 max,编译程序会进行相应的实例化。

C ++ 中的模板是类型安全的,编译程序在编译时刻进行类型检查。例如,对于 max 的例子,T 不能被实例化为某个自定义的类,除非这个类也支持运算符 <。从另一个角度看,模板也规定了类型参数必须满足的要求:实在参数必须能够支持特定的操作,否则编译程序会报出类型错误。当然,这样的要求不能给保证实例化得到的函数就是正确的。比如,标准模板库中的函数模板 sort( ) 要求元素类型支持运算符 <。但是,实际上只有当这个运算符代表了一个全序时,sort 才能够给出正确的结果。

第二种模板称为类模板。类模板可以看作是参数化的类定义。只要我们给出其中的参数,即可实例化得到相应的类,这种模板通常被用来实现容器类。人们只需要指定元素类型,即可得到特定的类。比如对于 STL 中的列表模板 list, list <int> 表示整数表,而 list <string> 表示字符串的表。我们可以直接使用它们来说明变量,比如

```
List <int> li;
List <string> ls;
```

类模板可以被部分实例化。当模板具有多个参数时,仅仅给出部分模板参数的取值可以得到另一个模板;另一种部分实例化虽然给出了所有参数的取值,这些值中仍然包含有参数。

在早期的 C 语言中,函数模板的功能是由宏实现的。和宏相比,模板可以避免很多错误,且使用于定义更大的函数。

模板的主要缺陷是编译出错信息难以理解。这常常导致初学者放弃使用模板。同时由于编译程序支持较差,使用模板的代码的易移植性也较低。

参考文献

ISO/IEC 14882. Information technology-Programming Languages—C ++ . 2011
（赵建华）

mohu fangzhen

模糊仿真(fuzzy simulation)　将模糊数学和仿真结合起来,以用于解决具有模糊不确定性系统的建模并进而实现仿真的技术(参见**计算机仿真**)。

模糊不确定性是指已经出现、但由于人们认识不清所包含的不确定性。模糊数学方法可用于模型信息与测量数据的不确定性建模,从而产生模糊模型。将用模糊模型(或含有模糊模型)描述的系统进行仿真时,需要建立模糊仿真模型,从而产生了相应的仿真技术。模糊仿真可分为模糊定量仿真和模糊定性仿真。

模糊定量仿真　模糊定量仿真是在传统定量仿真基础上,通过引入模糊集、模糊关系及模糊运算,增加对系统所具有的模糊不确定性成分的处理。

模糊仿真的研究对象是具有模糊不确定性的系统(简称模糊系统)。这种系统的模糊不确定性可以下述方式出现:部分输入具有模糊不确定性、部分参数具有模糊不确定性、部分初态具有模糊不确定性、部分结构具有模糊不确定性。上述任何一种模糊不确定性,均可能导致部分甚至全部状态序列和输出序列出现模糊不确定性。可引入模糊集和模

糊关系来表征其模糊不确定性。对于上述前三种模糊不确定性,可用一个模糊集来表征;对于结构的模糊不确定性,可用一个模糊关系来表征。

对模糊定量仿真而言,根据其系统中各种模糊不确定量的关系,既可以通过模糊运算进行,也可以通过常规实值运算进行。当采用模糊运算时,仿真过程中将直接用模糊值变量和参数进行一次仿真;当采用常规实值运算时,仿真时将针对模糊变量取值域中的抽样点或关键点分别进行。例如对三角形模糊数将分别进行三次仿真。常用的模糊仿真方法包括蒙特卡罗法(参见**蒙特卡罗仿真**)、不相关不确定法以及相关确定法等。

模糊定性仿真　定性理论中一般用模糊数学作为描述系统的手段。最初,采用区间模糊数来描述系统的定性的行为,英国的 Qiang Shen 进一步用凸模糊数来描述定性值,此后,引入概率论来度量生成的多个行为的可信度。

模糊定性仿真是在传统定性仿真(参见**定性仿真**)基础上,通过引入模糊集、模糊关系及模糊运算,以增加对具有模糊不确定性知识的定性系统的处理,所形成的一门新型仿真技术。

在传统定性仿真中,量空间是指变量的取值空间,通过一系列有序路标加以定义。在模糊定性仿真中,量空间通过有限个模糊数来定义,是由有限个模糊数所组成的集合。其中模糊数就是变量的可能取值,是经过对变量的实数取值范围做有限的、任意的划分而得到的。每一个模糊数都和一定的常识性或概念性含义,如“大”“小”“较大”“高”等相对应。在实际仿真过程中所得到的定性信息可通过最大隶属度原则或最大近似原则确定其定性值。1993 年英国的 Q. Shen 和 R. Leitch 提出模糊仿真方法 Fusim,较好地把定性仿真技术与模糊数学进行了集成,这种方法用模糊集合扩展了传统的量空间,从而增强了对函数关系的强度信息和变量的变化速率的顺序信息的描述。

模糊仿真方法也存在一些弱点,比如很难确定系统真实值与模糊量空间的映射,并且模糊量值及其空间一旦确定后就不再变动,不能根据需要引入新的模糊量,限制了模糊模型的描述能力。

参考文献

1. 廖良才,等. 模糊仿真理论综述及探讨. 模糊系统与数学,2000,14(3)

2. 贾利民,等. 模糊仿真——定性模型的建立与辨识. 计算机仿真,1994,(1)

(王威)

mohu luoji

模糊逻辑(fuzzy logic)　研究不确定性(特别是模糊性)推理与知识表示的逻辑基础及其应用的学科。它正处于发展之中,目前主要包括三个方面:狭义模糊逻辑、广义模糊逻辑与模糊(近似)推理。

狭义模糊逻辑是指真值集为单位区间的连续值逻辑或其他一些可能应用于模糊性处理的**多值逻辑**,它们在语形上与经典逻辑没有本质区别,只是真值集被扩充,基本上不依赖于模糊集理论。对于狭义模糊逻辑的研究,最早可以追溯到 J. Lukasiewicz 20 世纪 20 年代的工作,相继有了较为丰富的内容。特别是 J. B. Rosser 与 A. R. Turquette 在 1952 年就作为未决问题提出了建立超出一阶谓词演算的多值逻辑的设想,并提及多值集的概念,这实际上就是模糊集。到了 1965 年,L. A. Zadeh 在不确定性信息处理的客观背景下,独立、系统地建立了模糊集理论并指出其在人工智能、自动控制、管理与决策等领域中的应用。正由于此,连续值逻辑及一些相关的领域重新引起了人们的兴趣,一些学者甚至将其改称为模糊逻辑。反过来,狭义模糊逻辑又是模糊集理论乃至模糊数学之逻辑基础;模糊数学的发展是对上述 Rosser-Turquette 问题的部分解答。在模糊集理论出现之后,有关狭义模糊逻辑的工作主要有:①模糊逻辑的归结原理及其在完全分配格上的推广;②模糊逻辑函数的极小化、分析与综合、险态检测及模糊逻辑的函数完备性;③直觉主义模糊逻辑、集合论的形式化系统;④模糊模态逻辑;⑤连续值与剩余格值逻辑的语义方法在不分明拓扑中的应用。

由于狭义模糊逻辑的表达能力有限,不足以处理不确定性推理与知识表示中的许多问题,一些学者致力于发展能刻画语言真值、广义量词、修饰词、限定词,允许假设与推理规则都具有不确定性,可以作近似证明(推理)的广义模糊逻辑系统,试图为不确定性的处理提供较为完整的逻辑框架。现阶段广义模糊逻辑主要沿下述两个方向独立地发展,但离建立能统一这两个方面工作的完整形式化演算还有相当的距离。

(1) 20 世纪 70 年代以来 L. A. Zadeh 等以模糊集为工具提出了一种对语言真值、模糊量词、修饰词与限定词作语义解释的翻译规则。①语言真值被解释为单位区间内的模糊数,作为其简化,有时也考虑单位区间的子区间。②模糊量词通常利用模糊集的基数刻画,此即比例法。这种方法虽然十分自然,但逻辑性质很不理想。另一种局部化地刻画模糊量词

的方法是代换法,它只适用于有限论域。为了克服这些困难,有的学者提出了用 Sugeno 模糊测度与积分刻画模糊量化命题的方法。③模糊修饰词是用来表现语气副词的,它适度地改变命题的意义,通常以平移或指数函数刻画。④限定词有真值、概率与可能性限定三种类型。1985 年刘叙华建立了算子模糊逻辑,其原始目的可以看作处理限定词的一种简化的形式化方法,后在不同的直观意义下产生了多种变形。目前,这方面的工作还是初步的;除了刘叙华已经证明算子模糊逻辑中归结原理的完备性外,对于带有语言真值、模糊量词、修饰词、限定词的广义模糊逻辑的完全形式化还有许多困难的问题有待解决。由于在自然语言理解等方面的潜在应用价值,语言真值、模糊量词、修饰词与限定词的数学处理及其与 Keisler 概率量词等的关系值得深入研究。

(2) 1979 年 J. Pavelka 提出了假设带有不确定性的希尔伯特型完备格值演绎系统,并在有限链与连续值情形获得了有关相应命题逻辑的完备性与不完备性的结果。其后,人们证明了对应的一阶模糊逻辑的完备性;将 Pavelka 的希尔伯特型演绎系统推广到 Gentzen 型模糊自然演绎或推理规则具有不确定性的情形;建立了允许假设与推理规则作近似匹配的数值化近似证明理论,并证明了带有这种近似证明(但真值仍取二值的)命题与一阶逻辑的完备性。这方面仍有许多问题需要讨论,如各种不确定性的组合机制与推理过程中推理规则按重要性程度的排序。这些问题的解决可能为人工智能系统的基本积木块——产生式系统中不确定性处理奠定较为完整的逻辑基础。

将模糊逻辑应用于智能化电器产品的开发与工业过程控制,这些实际应用的理论基础是模糊推理,它与狭义、广义模糊逻辑之间没有太多的本质联系。1972 年,L. A. Zadeh 提出了模糊推理的关系合成规则,现有的绝大多数模糊推理方法都是关系合成规则的变形(如真值限定方法)或扩充(如多维、多条件及区间值模糊推理)。20 世纪 70 年代后期,E. H. Mamdani 等将模糊推理方法应用于模糊控制器的设计,这为 20 世纪 80 年代后期模糊技术的兴起打下了基础。近年来,模糊逻辑与神经网络的结合受到了人们很大的重视,它反映了人工智能中符号主义与连接主义机制的综合集成;带模糊量词与修饰词的模糊推理、模糊推理在类比推理中的应用、模糊推理的摄动及其对模糊控制器稳定性的影响也开始引起人们的注意。设 U,V 是两个论域,x,y 分别是取值于 U,V 的变元,A,A' 是 U 的模糊子集且 B 是 V 的模糊子集。模糊推理的基本模式是

Ant.　If x is A then y is B

　　　x is A'

Cons.　　　y is $B' \triangleq C(A,B;A') = ?$

这里结论 $B' = C(A,B;A')$ 是 V 的模糊子集,它通常由关系合成规则给出如下:

$$C(A,B;A')(y) = \sup_{x\in U} A'(x)T(A(x)\rightarrow B(y)),\ y\in V$$

其中 T 是合取算子,→是蕴涵算子,由于结论 $B' = C(A,B;A')$ 依赖于算子 T,→的选取,在应用中如何根据实际问题恰当地选取合取与蕴涵算子往往对于应用效果产生决定性的影响;许多学者已经对一些常用的合取与蕴涵算子分析比较了模糊推理的结果。在理论上,因为关系合成方法不满足分离规则 $C(A,B;A) = A$,人们通常认为这是模糊推理的不合理性。一方面,一些学者利用模糊关系方程的理论考虑了对于什么合取与蕴涵算子、在什么情况下分离规则成立的问题;另一方面,可以证明对于绝大多数常用的合取与蕴涵算子,每个模糊推理模式等价于另一个满足分离规则的模式,从而揭示了其隐藏在表面不合理性之后的深刻的合理性。

参考文献

1. Zadeh L A. Fuzzy sets and applications. New York:John Wiley & Sous,1987

2. Pavelka J. On fuzzy logic Ⅰ,Ⅱ,Ⅲ. Zeitschr. f. math. Logik und Grundlagend. Math.,1979,25(1):45-52,(2):119-134,(5):447-464

3. 刘叙华. 基于归结方法的自动推理. 北京:科学出版社,1993

(应明生)

mohu shujuku

模糊数据库(fuzzy database)　能够处理模糊数据的数据库,模糊数据的表示主要有模糊区间数、模糊中心数、模糊集合数和隶属函数等。

模糊数据库的研究开始于 20 世纪 80 年代初,旨在克服传统数据库难以表达和处理模糊数据的弱点,进而扩展数据库的功能。迄今为止,研究人员在模糊关系数据库方面开展了大量研究工作,对关系数据库进行模糊扩展。首个模糊关系数据库是 FRDB。其他一些模型也相继被提出,如 Buckles-Petry 模型、Prade-Testemale 模型、Umano-Fukami 模型和 GEFRED 模型。近年来,亦有许多工作对关系模

型之外的其他数据库模型进行模糊扩展，如模糊实体-关系模型、模糊面向对象数据模型等。

模糊数据库尚处于研究阶段，其研究内容主要包括三个方面：

(1) 模糊数据的表示　在传统的关系数据模型中引入模糊性，使之能够表示不确定或不精确的信息，主要表示方法有模糊区间数据、模糊中心数据、模糊集合数据和隶属函数等。

(2) 模糊查询　通过使用模糊集合或模糊谓词对模糊数据库进行查询，即查询条件可包含模糊集合或模糊谓词。模糊数据库查询语言中比较突出的有，由 P. Bosc 等人提出的 SQLf 和由 J. Galindo 等人提出的 FSQL。这些语言能够在 SQL 语句中表示模糊性，例如模糊条件、模糊比较器、模糊常量、模糊约束、模糊阈值、语言标签等。

(3) 模糊关系数据库的设计　和传统关系数据库一样，模糊关系数据库的设计旨在获得合理的数据库模式，进而避免可能出现的数据冗余和修改异常。

参考文献

Galindo J, Urrutia A, Piattini M. Fuzzy databases: modeling, design and implementation. Idea Group Inc., 2006

（李建中　邹兆年）

mokuaihua fangfa

模块化方法（modular method）　一种软件开发方法，把一个待开发的软件分解成若干小的简单的部分，称为**模块**。每一个模块都独立地开发、测试，最后再组装出整个软件。这种开发方法是对待复杂事物的“分而治之”的一般原则在软件开发领域的具体体现。

模块化开发方法涉及的主要问题是：模块设计的规则，系统如何分解成模块。

模块是执行一个特殊任务或实现一个特殊的抽象数据类型的一组例程和数据结构。模块通常由两部分组成。接口：列出可由其他模块或例程访问的常数、数据类型、变量、函数等；实现：私有量（只能由本模块自己使用的）及实际实现本模块的源程序代码。

模块的接口部分刻画了各个模块是如何耦合的，是其他模块的设计者和使用者所需要知道的。而实现部分是各模块的内部事务，其他模块并不需要知道。这也体现了对待复杂事物的另一原则——抽象原则（在软件领域称为信息隐蔽原则），即把非本质的性质隐藏起来，只突出那些本质的性质，以减轻人们思考和注意的负担。模块化澄清和规范化了软件中各部分间的界面。如此就便利了成组的软件设计人员工作，也促使了更可靠的软件设计实践。

在把系统分解成模块时，应该遵循以下的规则：①得到最高的模块内聚，也就是在一个模块内部的元素最大程度地关联。只实现一种功能的模块是具有最高内聚的，具有三种以上功能的模块则是低内聚的。②最低的耦合，也就是不同模块之间的关系尽可能弱。③模块大小适度。④模块调用链的深度（嵌套层次）不可过多。⑤接口干净，信息隐蔽。⑥尽可能地复用已有模块。

如何对一个规约进行分解，以得到模块化的系统结构。已经有一些基于设计规则的方法。

(1) 数据结构设计方法　最著名的是 Jackson 结构化设计（参见 **Jackson 系统开发方法**）。它从画出所有输入/输出数据的逻辑结构图开始，最后得到程序结构图，反映了系统结构。可能还需要继续对其中模块求精，得到更低级的模块，但是基本程序结构是不变的。

(2) 功能分解　步骤是：陈述出功能意图（即要解决的问题），进行功能求精（即划分层次），连接求精了的功能，进行检查，再求精，再检查，直至得到满意的解决为止。这种方法，也称为结构化设计、层次化分解、模块分解、功能分解。

(3) 数据流设计　步骤是：把问题分解成由动作图（也称为进程）和数据图（也称为流）组成的数据流图，还有存放待处理的静态信息的存储元素。然后从数据流图中找出中心进程，以它为根，把数据流图转换为树形结构。按照功能分解形式来分解进程，即把模块分为三类（输入、变换、输出），进行进程求精，得到 PDL 语言表示。最后把 PDL 变成某种程序语言。

(4) 面向对象的设计　这一方法要求标识出对象及其属性、每个对象所需要的操作。把数据及函数封装在一起，以形成类。并建立其间相互可见的关系，即许可的调用与被调用关系，形成每个类的界面。最后是实现每个类（参见**面向对象方法**）。

模块化方法的优点是明显的，难点所在是如何处理大型问题。分而治之的原则是好的，然而对大型问题会难以找到下手之处。当从一个角度看问题时，要求隐蔽的信息是这些，而换一个角度看时，却要求隐蔽另外的信息。模块化方法在发展的早期主要是手工设计方法，以后发展了许多自动化工具来支持它们。最好是几种方法组合使用，各自发挥所长，但是内部表示和工具间的相容性是一大困难。

参考文献

Lewis T G. CASE: computer-aided software engineering. New York: Van Nostrand Reinhold, 1991

(董榅美)

mokuai jiegou tu

模块结构图(modular structure diagram) 一种表示软件各模块之间控制关系的图形工具。如图1所示。

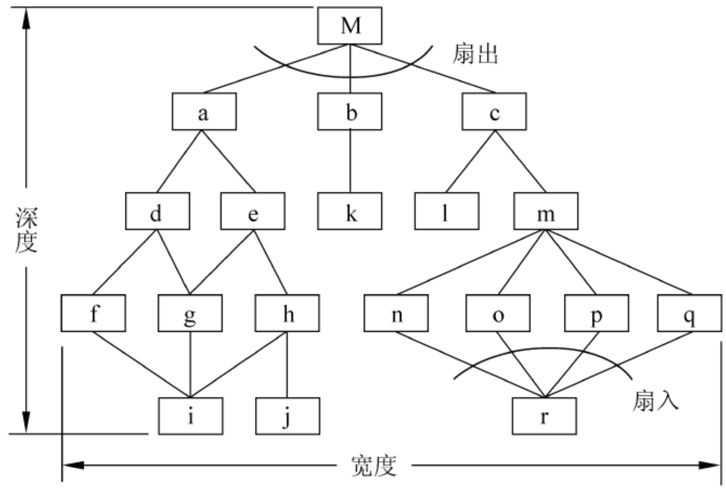


图1　模块结构图

其中,矩形表示一个软件模块,矩形之间的线段表示模块之间的控制关系。按控制关系,可以把模块分为若干个层次。上一层次的模块称为控制模块,而下一层次的模块称为从属模块。模块结构图的度量属性,主要包括模块结构图的宽度、深度以及模块的扇入和扇出。模块结构图的宽度是指具有最多模块的那一层之模块数目,或称模块结构图的控制跨度;模块结构图的深度是指模块控制层的数目,或称模块结构图的控制深度;模块的扇入是指其控制模块的数目;模块的扇出是指其从属模块的数目。

在实践中,模块结构图通常用于软件系统概要设计、标识系统的模块结构。　(李宣东　王立福)

moni jisuanji

模拟计算机(analog computer) 可对连续物理变量进行数学运算的解算装置。当把模拟计算机的各种不同功能部件按照一个物理系统的数学描述连接时,就构成这个物理系统的数学模型,可用来直接模仿该物理系统的行为。按模拟计算机中代表变量的物理量性质和运算部件构造的不同,模拟计算机可分为机械式、机电式或电子式等不同类型。在模拟计算机中,加、减、乘、除、微分、积分等数学运算,皆由相应的运算部件,如加法器、乘法器、积分器等并行完成。按运算的性质,运算部件可分成线性和非线性两大类。

在电子模拟计算机中,变量为连续变化的直流电压、电流或者电荷。各种运算部件主要是由运算放大器和精密电阻、电容以及特殊的开关元器件等构成。运算放大器是构成各种运算部件的核心器件,它由一个高放大倍数的直流放大器A和输入阻抗Z_i以及跨接于输出与输入之间的反馈阻抗Z_f组成,如图1所示。按基尔霍夫定律和欧姆定律,可得出其回路方程如下:

$$\frac{E_o - V}{Z_f} + \frac{E_i - V}{Z_i} = I$$

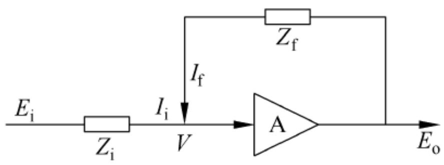


图1　运算放大器框图

式中,E_i,E_o分别为输入、输出电压,I为电流。当直流放大器的放大倍数很大($>10^7$),Z_i及Z_f也足够大时,$E_o \gg V$, $E_i \gg V$, $I \approx 0$。于是可得出模拟计算的基本关系

$$\frac{E_o}{E_i} = -\frac{Z_f}{Z_i}; \qquad E_o = -\frac{Z_f}{Z_i}E_i$$

若阻抗Z_i, Z_f皆为纯电阻,令为R_i,R_f, 且$R_i =$

R_f，便构成一个反相器；一般情况下，$R_i \neq R_f$，便可构成一个比例运算器。若 Z_i 是多个电阻的并联［图 2(a)］，可导出如下关系：

$$E_o = -\left(\frac{R_f}{R_1}E_1 + \frac{R_f}{R_2}E_2 + \cdots + \frac{R_f}{R_n}E_n\right)$$

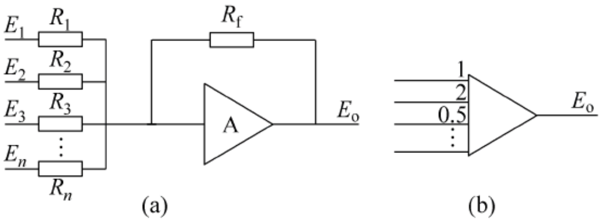


图 2　比例加法器框图及符号图

这就构成一个比例加法器；若 Z_f 为一个电容器［图 3(a)］，便可得出以下动态回路方程：

$$V_o = \frac{q}{C} = -\frac{1}{C}\int_0^t I\mathrm{d}t = -\frac{1}{RC}\int_0^t V_i\mathrm{d}t$$

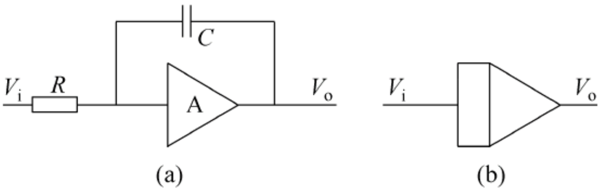


图 3　积分器框图及符号图

可见输出电压 V_o 是输入电压 V_i 对时间的积分。积分的速度取决于时间常数 RC。这就构成了积分器，它是模拟计算机解常微分方程的最主要的基本运算部件。在图 2(b)和图 3(b)中，还分别给出了加法器和积分器的符号图。标注在加法器输入端的数字表示比例系数(R_f/R_i)。积分器也可以同时实现加法运算，图 4 是具有加法功能的积分器的符号图。

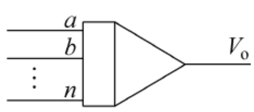


图 4　具有加法功能的积分器符号图

模拟计算机的求解方法可以用一个例子来说明。例如对下面的微分方程求解

$$\frac{\mathrm{d}^2x}{\mathrm{d}t^2} + a\frac{\mathrm{d}x}{\mathrm{d}t} + bx = c$$

若不考虑初始条件，上式可进一步写成

$$\frac{\mathrm{d}x}{\mathrm{d}t} = \int_0^t\left(-a\frac{\mathrm{d}x}{\mathrm{d}t} - bx + c\right)\mathrm{d}t$$

求解时，通过排题板把反向器和积分器与上式对应地按图 5 所示连接起来，就得到了 x，即该微分方程的解。如果上述微分方程是对某一真实系统的描述，则图 5 就是该系统的模拟。

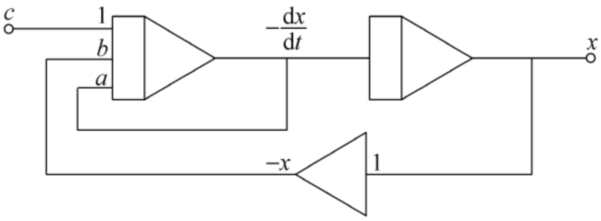


图 5　一个真实系统的模拟

利用特殊设计的函数发生器与运算放大器、比例加法器等相配合，既能产生各种非线性函数，也能实现如乘法、乘方、开方、除法等基本算术运算。例如，用两个平方函数发生器和 3 个加法器，可以构成一个乘法器，如图 6(a)所示。图中 FG 代表函数发

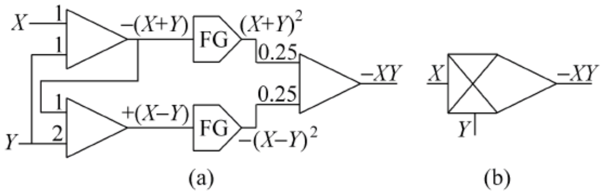


图 6　四分之一平方乘法器框图及符号图

生器，其输出值为输入值的平方而符号相反(亦可相同)。作为输出的比例加法器，采用 1∶4(图中用 0.25 表示)的比例系数，所以，这种乘法器亦称四分之一平方乘法器，如图 6(b)所示。于是最终输出为

$$E_o = \frac{1}{4}[-(X+Y)^2 + (X-Y)^2] = -XY$$

模拟计算机除运算部件外，还有接线排题板和相应的控制电路，用以连接各种运算部件并控制机器的启、停和其他操作。由于模拟计算机工作的连续性、并行性和实时性，而且操作简便，十分适用连续系统的实时仿真。其主要缺点是受元器件精度限制和运算放大器零点漂移的影响，整机的运算精度远低于数字计算机。可以把模拟计算机与数字计算机结合起来，组成混合计算机。

参考文献

1. Korn G A, Korn T M. Electronic analog computers. New York: McGraw-Hill, 1956

2. Frederiksen T M. Intuitive analog electronics. New York: McGraw-Hill, 1989

(李人厚　童頫)

moni shuru shuchu tongdao

模拟输入输出通道（analog input/output channel）　计算机为输入模拟信号和输出模拟信号而设的通道。在工业控制系统中的仪器设备，检测或控制的对象常是连续变化的物理量，如温度、压力、电压、电流等。这些连续变化的信号必须转换成数字信号后计算机才能接收和处理。计算机发出的控制命令，常常需要把它转换成模拟信号，才能驱动执行机构。工业控制系统的结构如图1所示。由设备到计算机系统的通道就是模拟输入通道。由计算机系统到设备的通道就是模拟输出通道。

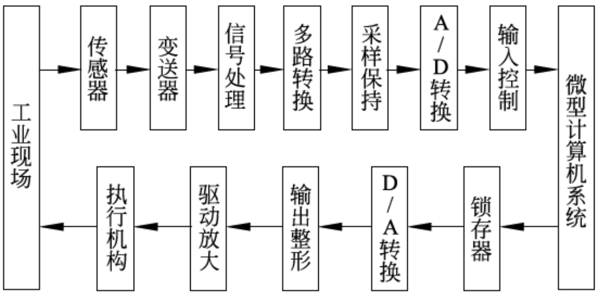
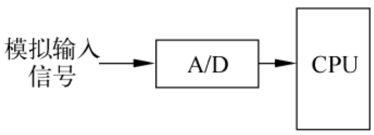


图1　工业控制系统的结构

模拟输入通道由采样保持器、多路开关和模数转换器等组成。它有下面几种结构形式：

1. 不带采样保持器的单通道（图2）

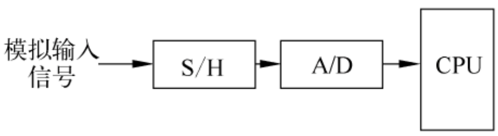


CPU——中央处理器
A/D——模数转换器

图2　不带采样保持器的单通道

这种结构常用于低频或直流信号。

2. 带采样保持器的单通道（图3）

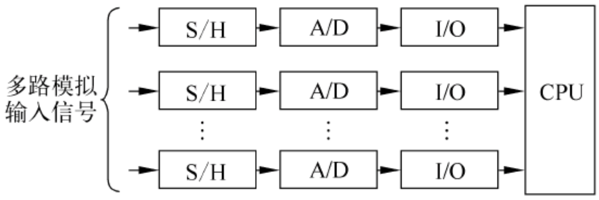


S/H——采样保持器

图3　带采样保持器的单通道

适用于模拟信号时间变化率较大的情况。与上一种方式不同的是增加了采样保持器S/H。

3. 每路有单独采样保持器的多通道（图4）



I/O——输入输出电路

图4　多通道模数转换

这种结构用于高速系统，各通道独立进行转换，互不干扰，缺点是所需硬件多。除了采样保持器及模数转换器外，每个通道要增加输入输出接口电路。

4. 多路共享模数转换器的通道（图5）

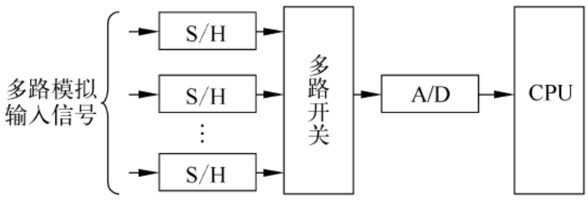


图5　多通道共享模数转换器

这种结构的速度比上一种慢，因为信号要经过多路开关进行转换，串行进入模数转换器，但各路采样保持器是独立的。

5. 多路共享采样保持器的通道（图6）

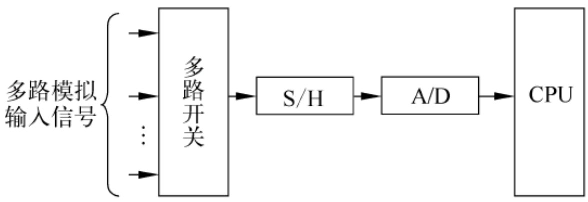


图6　多通道共享采样保持器与模数转换器

这种结构硬件最省。

模拟输出通道除了单通道形式外，也有多通道的形式。

模拟输入输出通道中核心器件是**模数转换器**和**数模转换器**。

传感器将控制对象的温度、压力、流量、位移等物理量检测出来并转化成一定范围内连续变化的电流或电压模拟量，再通过模数转换器转换成二进制或十进制的数字量输入到计算机。模数转换器电路已有专用的芯片，采用的转换原理常见的有逐次逼近式和双积分式。逐次逼近式转换精度和速度都比

较高,是12位以下模数转换器应用最广的一种。双积分式抗干扰能力强,转换精度高,但转换速度慢,因此多用于数字式测量仪表中。

数模转换器把数字信号转换成连续的用电压或电流表示的模拟量。它把数字量作为输入量,实时产生与数字量相对应的模拟量。转换方式多采用T形电阻解码网络。数字量的输入可以是并行的也可以是串行的。输出量可以是电压也可以是电流。

模数转换器和数模转换器也广泛应用于仪表和外围设备中。例如,形成音乐图像的音频视频信号经模数转换成数字信号记录到光盘。从光盘中读出的数字信号经数模转换成音频或视频信号,重现音乐或图像。

随着集成电路技术的进步,模数转换器、数模转换器、采样保持器、多路开关等电路都有芯片供应。

参考文献

1. 张明,李训涛. 计算机测控技术. 2版. 北京:国防工业出版社,2010

2. 秦鹏,等. 微机接口技术实用教程. 2版. 北京:清华大学出版社,2009 (林兼)

moshi fenleiqi

模式分类器(pattern classifier) 对用以描述或表示事物和现象的数据进行分类的装置或过程。它是**模式识别**学科的重要组成部分。

智能行为的一个重要表现就是能够根据实际应用提出的要求,正确地把被识别对象划分到某一类别或某一范畴中去。模式分类正是在模式表示已经确定的情况下实现这种智能行为的机器方法。

为了使分类器具有良好的分类性能,首先要对分类器进行训练。最常见的一种训练方式是用一组特定的样本数据(观察)连同它所属的类别(称为训练样本集)作为输入,在一定的规则下,求出分类器的结构参数(训练阶段)。通过训练,从而具有对未知类别的模式进行分类的能力(工作阶段)。这种训练过程又称作**监督学习**(参见**人工神经网络**)。与此相对应的是非监督学习(参见**非监督学习**)。

用监督学习实现分类器设计可分为Bayes决策方法和判别式方法。

Bayes决策方法的基础是用同一类别的训练样本集估计该类的概率密度函数$p(x|w_i)$。其中$w_i=1,2,\cdots,C$,表示C个类别,x是表示模式的随机特征向量。如果待识别的模式的特征向量为X,根据Bayes公式,可求得X的后验概率

$$P(w_i \mid X) = \frac{p(X \mid w_i)P(w_i)}{\sum_{w_i=1}^{c} p(X \mid w_i)P(w_i)}$$

其中$P(w_i)$是类别为w_i的先验概率。Bayes决策方法是比较各个类的$P(w_i|X)$,具有最大值的w_i即为该模式所属的类别。理论上可以证明,Bayes决策方法具有最小的误识率,因此是一种最优分类器。Bayes决策方法需要预先确定各个类的先验概率和类概率密度函数。由于概率密度函数的估计是一个很困难的问题,因此在很多实际问题中,实现Bayes分类器有较大的困难。由此,人们转而开始研究下面这些直接从样本设计分类器的方法。

判别式方法则直接求取分界面函数,其基本思路是根据对问题性质的分析,首先确定分界面的函数类(例如线性函数、二次多项式等),然后在一定的准则下(例如训练集中样本被错分的数量最少),求出其中的最优函数作为分界面,从而把整个特征空间划分为若干区域,落入到某个区域中的模式就分类为该区域所对应的类别。函数类的选择在很大程度上还取决于特征空间的维数和训练样本集的样本数。当维数较高,样本数又不多的情况下,为了使分类器有较好的推广能力,通常采用线性函数,其相应的分界面则是超平面。例如:支持向量机方法(参见**支持向量机**)

最近邻规则是对训练集中的样本进行筛选后,直接存入计算机。计算被识别模式与这些样本的距离,与被识别模式距离最小的样本所具有的类别即为被识别模式的类别。理论上证明,当训练集样本数接近无穷多时,这种分类方法的分类性能接近最优分类器。

当分类问题的特征不是用n维特征空间中的点(特征向量),而是用非度量特征(如:颜色,形状,性别等)表示时,以上很多方法无法应用。这时,可以采用决策树方法(参见**决策树**)。最常用的决策树方法是C4.5。

把多个分类器进行融合得到一个综合的分类器以提高分类的性能被称为多分类器方法,也被称为集成学习方法(参见**集成学习**)。常见的多分类器方法包括:Bagging,Adaboost,随机子空间方法。

参考文献

1. Sklansky J, Wassel G N. 模式分类器和可训练机器. 阎平凡,等译. 北京:科学出版社,1987

2. Duda R O, Hart P E, Stork D G. Pattern classi-

fication. 2nd ed. John Wiley & Sons,2001

3. 张学工. 模式识别. 3 版. 北京：清华大学出版社,2010

（边肇祺　张长水）

moshi shibie

模式识别（pattern recognition） 对表征事物或现象的各种形式的（数值的，文字的和逻辑关系的）信息进行处理和分析，以便对事物或现象进行描述、辨认、分类和解释的过程。它是信息科学和**人工智能**的重要组成部分。

英文“pattern”源于法文“patron”，本来是指可作为大家典范的理想的人，或用以模仿复制的完美的样品。在模式识别学科中“模式”具有更广泛的意义。人们在观察事物或现象的时候，常常要寻找它与其他事物或现象的相同或不同之处，根据一定的目的把并不完全相同的事物或现象组成为一类。字符识别就是一个典型的例子。例如汉字“中”可以有各种写法，但都属于同一类别。更为重要的是，即使对于某个“中”的具体写法从未见过，也能把它分到“中”这一类别。在以上例子中，模式是和类别（集合）的概念分不开的，只要认识这个集合中的有限数量的事物或现象，就可以识别出属于这个集合的任意多的事物或现象。为了强调能从具体的事物或现象推断出总体，我们就把个别的事物或现象称作“模式”，而把总体称作类别或范畴。也有的学者认为应该把整个的类别称作模式，这样的模式是一种抽象化的概念，如“房屋”，“铁路”，“通俗音乐”等等都是模式，而把具体的对象，如人民大会堂称作“房屋”这类模式中的一个样本。这种名词上的不同含义是容易从上下文中弄清楚的。

模式还可分成是抽象的模式和具体的模式。前者如意识、思想、议论等，属于概念识别研究的范畴，是人工智能的另一研究分支。后者是可以用物理的、化学的、生物的传感器对对象进行测量得到的声音、图片、文字、符号等数据。

模式识别研究主要集中在两方面，即研究生物体（包括人）是如何感知对象的，属于认知科学的范畴，以及在给定的任务下，如何用计算机实现模式识别的理论和方法。前者是生理学家、心理学家、生物学家和神经生理学家的研究内容，后者通过数学家、信息学专家和计算机科学工作者近几十年来的努力，已经取得了系统的研究成果。

早期的计算机模式识别研究着重在模型的建立上。20 世纪 50 年代末，一种简化的模拟人脑进行识别的数学模型——**感知机**被提出，初步实现了通过给定类别的样本对识别系统进行训练，从而能对其他未知类别的模式进行正确分类。60 年代提出了一种基于基元关系的句法识别方法，傅京孙在这个领域进行了卓有成效的工作，形成了句法模式识别的系统理论（参见**句法模式识别方法**），另外，用统计决策理论求解模式识别问题得到了迅速的发展。70 年代前后出版了一系列反映统计模式识别理论和方法的专著（参见**模式分类器**）。80 年代，人工神经网络的研究为模式识别技术提出了一种新的途径（参见**人工神经网络在模式识别中的应用**）。90 年代提出了支持向量方法（参见**支持向量机**）和 Adaboost 方法（参见**集成学习**）使得模式识别有了很大发展。

模式识别与**机器学习**密不可分。**计算学习理论**为模式识别奠定了理论基础。

一个计算机模式识别系统基本上是由三部分组成的，即数据采集、数据处理和分类决策或模型匹配。数据采集是指通过各种传感器把被研究对象的各种物理变量转换为计算机可以接受的数值或符号（串）集合。习惯上，称这种数值或符号（串）所组成的空间为模式空间。之后要进行的数据处理是为了从这些数值或符号（串）中抽取出对识别有效的信息，其中包括消除噪声，排除不相干的信号以及与对象的性质和采用的识别方法密切相关的特征的计算（如表征物体的形状、周长、面积等）以及必要的变换（如为得到信号功率谱所进行的快速傅里叶变换）等。然后通过特征选择（参见**特征选择**）和提取或基元选择（参见**句法模式识别方法**）形成模式的特征空间。最后的模式分类或模型匹配就在特征空间的基础上进行。系统的输出或者是对象所属的类型，或者是模型数据库中与对象最相似的模型编号。

模式识别已经在生物、医学、天气预报、工业产品检测、字符识别、语音、图像、文字的识别和分类等许多方面得到了成功的应用，例如：指纹图像识别，手写体文字识别。所有这些应用都是和问题的性质密不可分的。理论研究结果表明，不存在对所有模式识别问题都适用的单一模型和解决识别问题的单一技术，但是可以在具体的应用领域发展出有效的算法。

参考文献

1. Watanabe S. Pattern recognition：Human and mechanical. New York：Wiley，1985

2. 张学工. 模式识别. 3 版. 北京：清华大学

出版社，2010 （边肇棋 张长水）

motai luoji

模态逻辑（modal logic） 在非经典逻辑中，考察“必然”“可能”“偶然”等模态概念的逻辑性质，研究模态命题之间、模态命题与非模态命题之间形式推理的一个分支学科。模态逻辑的内容包括模态逻辑系统（不同的系统体现了对“必然性”概念的不同理解），语义理论，模态逻辑系统的元逻辑特性（例如：一致性、可靠性、完全性和可判定性），以及模态逻辑与现代逻辑其他分支学科的相互关系等。

模态命题逻辑系统 最基本的模态命题逻辑系统是 T，S_4，B 和 S_5。其中系统 T 可以用如下的方式表述：

（1）初始符号 ①p_1，p_2，…；它们可解释为可数个命题变元。②¬，→，L，（，）；它们可分别解释为否定词、实质蕴涵、必然算子、左括号和右括号。

（2）形成规则 ①每一个命题变元 p_i（$i=1,2,\cdots$）是公式；②若 A，B 是公式，则 ¬A、(A→B) 和 LA 是公式。

（3）公理（A，B，C 是任意的公式） ①$A\to(B\to A)$；②$(A\to(B\to C))\to((A\to B)\to(A\to C))$；③$(\neg A\to\neg B)\to(B\to A)$；④$\mathrm{L}A\to A$；⑤$\mathrm{L}(A\to B)\to(\mathrm{L}A\to\mathrm{L}B)$。

（4）推理规则（A，B 是任意的公式） ①分离规则：由 A 和 $A\to B$ 可推出 B。②必然规则：由 A 可推出 LA。

在系统 T 的基础上，再添加公理 $\mathrm{L}A\to\mathrm{LL}A$ 即构成系统 S_4。对系统 T 分别添加公理 $\mathrm{M}\,\mathrm{L}A\to A$ 和 $\mathrm{M}\,A\to\mathrm{L}\,\mathrm{M}A$（M为“可能算子”，其定义为 ¬L¬）则分别构成系统 B 和系统 S_5。这 4 个系统都是正规系统。系统 T 中没有任何归约律，S_4 中有两种归约律 $\mathrm{L}A\leftrightarrow\mathrm{L\,L}A$ 和 $\mathrm{M}\,A\leftrightarrow\mathrm{M\,M}A$，$S_5$ 中则有全部 4 种归约律成立（另两种是 $\mathrm{M}A\leftrightarrow\mathrm{L\,M}A$ 和 $\mathrm{L}A\leftrightarrow\mathrm{ML}A$）。因而在 S_5 中任何高于 1 级的公式都可以归约为 1 级公式。对系统 B 可以作某种直觉主义的解释，即把 ¬M解释为直觉主义否定。

在非正规系统中，受到逻辑学家关注的有 $S_{0.5}$，S_2，S_3，$S_{3.5}$ 等。

模态谓词逻辑系统 最基本的一阶模态谓词逻辑系统是 $T+BF$ 和 $LPC+T$，它们都是在模态命题逻辑系统 T 和经典的一阶谓词逻辑系统 LPC 的基础上建立起来的。其中系统 $T+BF$ 可以用如下的方式表述：

（1）初始符号 ①x_1，x_2，…；它们可解释为可数个个体变元。②A_1^1，A_1^2，…；A_2^1，A_2^2，…；…；…它们可解释为谓词字母，A_i^j 是 i 元谓词。③¬，→，L，∀，，，（，）；其中∀是全称量词。

（2）形成规则 ①$A_i^j(t_1,t_2,\cdots,t_i)$ （$i=1,2,\cdots;j=1,2,\cdots$）是公式，其中 t_1，t_2，…，t_i 是任意的个体变元。②若 A，B 是公式，则 ¬A，(A→B)，LA 和 $(\forall x_i)A$ （$i=1,2,\cdots$）是公式。

（3）公理（A，B，C 是任意的公式） ①～⑤形式同系统 T 的公理①～⑤。⑥ $(\forall x_i)A\to A(x_j)$，其中 x_j 对于 A 中的 x_i 是自由的，用 x_j 取代 A 中 x_i 的每一次自由出现所得的公式是 $A(x_j)$。⑦ $(\forall x_i)(A\to B)\to(A\to(\forall x_i)B)$，其中 x_i 在 A 中没有自由的出现。⑧ $(\forall x_i)\mathrm{L}A\to\mathrm{L}(\forall x_i)A$。

（4）推理规则（A，B 是任意的公式） ①和②同系统 T 的分离规则和必然规则。③ 概括规则：由 A 可推出 $(\forall x_i)A$（$i=1,2,\cdots$）。

系统 $T+BF$ 中的公理⑧称为巴坎公式。若在系统 $T+BF$ 的公理中去除巴坎公式，即构成系统 $LPC+T$。巴坎公式亦可表述为 $(\forall x_i)\mathrm{L}A\leftrightarrow\mathrm{L}(\forall x_i)A$ 或 $(\exists x_i)\mathrm{M}A\leftrightarrow\mathrm{M}(\exists x_i)A$，它体现了全称量词和必然算子，存在量词和可能算子的可交换性。包含巴坎公式的系统与不包含巴坎公式的系统有完全不同的模型特征。受到关注的其他一阶模态逻辑系统有 S_4+BF，$LPC+S_4$，$LPC+B$，$LPC+S_5$ 以及某些比 $LPC+T$ 更弱的模态谓词逻辑系统。

语义理论 主要有克里普克语义和代数语义，前者更为重要。模态命题逻辑系统的克里普克语义模型的一般形式为有序三元组，其中系统 T 的模型可以用如下方式表述：

一个 T 模型是一个有序三元组 $\langle W,R,V\rangle$，W 是非空集，称为可能世界集；R 是 W 上的一个二元自反关系，即 $R\subseteq W\times W$，并且对每一个 $w_i\in W$，都有 w_iRw_i；V 是赋值函数，其定义域是 $F\times W$，其中 F 为系统 T 中全体公式组成的集合，值域是 $\{1,0\}$，1 表示“真”、0 表示“假”，并且 V 满足以下 4 个条件：

（1）对每一个命题变元 p_j 和每一个可能世界 $w_i\in W$，有 $V(p_j,w_i)=1$ 或 $V(p_j,w_i)=0$，但不同时成立。

（2）对每一个公式 A 和每一个 $w_i\in W$，有 $V(\neg A,w_i)=1$，当且仅当 $V(A,w_i)=0$。

（3）对任意的公式 A，B 和每一个 $w_i\in W$，有 $V(A\to B,w_i)=1$，当且仅当 $V(A,w_i)=0$

或$V(B,w_i)=1$。

（4）对任意的公式 A 和每一个 $w_i \in W$，有 $V(\mathrm{L}A,w_i)=1$，当且仅当对每一个满足条件 w_iRw_j 的 $w_j \in W$，都有 $V(A,w_j)=1$。

当一个公式 A 在每一个 T 模型的每一个可能世界中都为真时，称公式 A 是 T 有效的。可以证明 A 为 T 有效的充分必要条件是 A 为 T 中的定理，即系统 T 具有可靠性和完全性。对 T 模型的二元关系 R 分别添加传递性（即若 w_iRw_j 且 w_jRw_k，必有 w_iRw_k）和对称性（即若 w_iRw_j，则有 w_jRw_i），则分别构成 S_4 模型和 B 模型。当 R 同时具有自反性、传递性和对称性时，为 S_5 模型。系统 S_4，B，S_5 都具有可靠性和完全性。借助于克里普克模型，还可以证明系统 T，S_4，B，S_5 都是能行可判定的。

模态谓词逻辑系统的克里普克模型要涉及到个体域 D。包含巴坎公式的系统 $T+BF$ 和 S_4+BF 的模型结构为有序四元组 $\langle W,R,D,V\rangle$，它的特点是：每个可能世界有着相同的对象域 D；然而，不同的可能世界中，对象的性质或对象间的相互关系则可以不同。而不包含巴坎公式的系统 $\mathrm{LPC}+T$ 和 $\mathrm{LPC}+S_4$ 的模型结构为有序五元组 $\langle W,R,D,Q,V\rangle$，其中 Q 是由 W 到 D 的幂集的一个函数，它为每一个可能世界 $w_i(\in W)$ 指定了一个对象域 $D_i(\subseteq D)$，即允许不同的可能世界有不同的对象域。

模态逻辑的研究成果促进了广义模态逻辑的时态逻辑、道义逻辑等分支学科的发展，克里普克的可能世界语义理论已被广泛应用于众多的现代逻辑分支学科以及自然语言逻辑的研究之中，模态逻辑在人工智能研究中的重要性也日益显现出来。

参考文献

1. Hughes G E，Cresswell M J. A new introduction to modal logic，London：Routledge，1996

2. Gabbay D，Guenthner F. Handbook of philosophical logic，Vol Ⅱ. Boston：D. Reidel Publishing Company，1984

（冯棉）

moxing jianyan

模型检验（model checking） 通过搜索待验证（软件或硬件）系统模型的有穷状态空间来检验该系统的行为是否具备预期性质的一种自动验证技术。

模型检验最初是由 E. M. Clarke 和 E. A. Emerson 在 1981 年提出的，他们设计了检验有穷状态迁移系统是否满足给定 CTL 公式的算法。J. P. Quielle 和 J. Sifakis 在同年也提出了类似的思想。在模型检验中，系统用有穷状态模型建模；其性质规约则通常是时序逻辑或模态逻辑公式，也可以用自动机语言描述；通过有效的搜索来检验有穷状态模型是否满足规约，如果不满足，它还能给出使性质公式为假的系统行为轨迹。1993 年 K. L. McMillan 基于 R. E. Byrant 的有序二叉判定图（OBDD）来有效地表示状态迁移系统，提出了符号化模型检验，大大提高了可有效应用模型检验技术的系统规模，使得模型检验在工业界逐步得到应用。模型检验已经成为形式化方法中系统验证的重要途径。

模型检验应用面临的主要问题是状态爆炸问题。由于系统的有穷状态模型的状态数量往往随其模型的并发分量的增加呈指数增长，因此，复杂系统建模时其可达的状态空间常常难于在计算机存储器中全部构建，也就无法进行模型检验了。一种途径是通过发现模型的状态空间的结构特点来缓解状态空间爆炸问题，主要的方法包括符号化模型检验、对称模型检验、偏序模型检验、On-the-fly 模型检验等。在符号化模型检验时，用布尔公式刻画状态集合和状态对集合，用 OBDD 来表示这些布尔公式，使用 OBDD 上的布尔操作来计算谓词转换子（其不动点刻画了 CTL 模态子），从而使模型检验在压缩了的符号化状态空间上来验证 CTL 性质。对称模型检验针对由多个完全类似的进程组成的系统，利用其模型的状态空间的对称性来生成压缩的且对模型检验等价的模型。偏序模型检验通过发掘系统中并发执行的迁移的交换性，减少本质上相同的状态，从而仅生成足以检验预期性质的小部分状态空间。On-the-fly 模型检验把状态空间生成和检验它是否满足性质合在一起进行，而不去预先生成整个状态空间，从而尽可能避免状态爆炸。另一种途径是通过抽象和分解把复杂系统的验证转化成模型检验可以处理的问题，主要的方法包括抽象方法、组合方法等。抽象方法通过去掉原来模型中与待验证性质无关的信息而获得简化模型的方式来减小模型检验时问题的规模。抽象必须满足：若简化的模型具备某性质，则原来的模型亦具备该性质。组合方法基于分而治之的思路来缩减模型检验时问题的规模，先验证系统构件的局部性质，然后把这些性质组合起来获得系统的全局性质。

模型检验在硬件设计和通信协议的形式验证上获得了巨大的成功。例如，利用模型检验技术，有效地发现了多处理器高速缓冲存储器一致性协议中的

一些错误,而这些错误在传统的模拟中往往未能发现。著名的基于模型检验的并发系统自动验证工具有 SMV 和 SPIN 等。2001 年 G. Holzmann 研发的模型检验系统 SPIN 获 ACM 软件系统奖。

模型检验技术仍在发展,主要方向是从系统的建模模型和性质规约语言的角度扩展模型检验,例如,模型检验解决实时和混合系统、软件系统的验证问题、一阶模态逻辑模型检验;与其他软件技术的结合,如模型检验与软件测试,与基于定理证明的形式验证技术的结合。

参考文献

1. McMillan K L. Model checking. Encyclopedia of Computer Science. 4th ed. Macmillan Reference Ltd, 2000

2. Clarke M, Schlingloff B-H. Model Checking. In: Robinson, Voronkov A, eds. Handbook of Automated Reasoning. Elsevier Science Publishers, 2001

3. 林惠民,张文辉. 模型检测:理论、方法与应用. 电子学报,2002,30(12A) (王戟)

moxinglun

模型论(model theory) 研究形式系统及其解释之间关系的理论。在 20 世纪 20 年代,T. Skolem 等人在数理逻辑研究中就已得到模型论性质的重要结果。但模型论形成系统的理论,大致在 20 世纪 50 年代,其奠基人应推 A. Tarski,A. Robinson 也作出了很大贡献。

一个形式语言的解释称为此语言的一个结构(模型)。一阶语言的结构是一个具有若干函数、关系和特指元素的集合(参见**一阶逻辑**),也称为泛代数。所以,模型论被形容为“泛代数+逻辑”。由于所涉及的逻辑系统不同,模型论又可分为:一阶模型论、高阶模型论、无穷长语言模型论、模态模型论、多值模型论等。由于在数理逻辑中以一阶逻辑发展最成熟,所以,模型论也以一阶模型论内容最丰富,应用最广泛。

如果一阶理论 T_1 的非逻辑公理集是一阶理论 T_2 的非逻辑公理集的有穷子集,则称 T_1 是 T_2 的有穷公理化部分。紧致性定理指出:如果一阶理论 T 的每个有穷公理化部分有模型,则 T 有模型。它的另一等价形式是:如果公式 A 在一阶理论 T 中有效,则 A 在 T 的某个有穷公理化部分中有效。紧致性定理的应用很广,可以用它讨论一阶逻辑的表达能力,证明某些概念不能用一阶逻辑表达。非标准分析的基础是实数系的非标准模型,其存在性的证明就使用了紧致性定理。

通常把论域的基数称为结构的基数,把一阶理论的公式集的基数称为该理论的基数。勒文海姆–斯拜伦定理指出,有模型的可数理论必有可数模型。后来,Tarski 将其推广为基数定理。设 λ 是无穷基数,如果基数为 λ 的理论有无穷模型,则它有基数为任何 $\alpha \geqslant \lambda$ 的模型。这个定理在模型论及公理集合论中常被使用。

不出现量词的公式称为开公式。设 U 和 B 是一阶语言 L 的结构。如果 U 的论域是 B 的论域的子集,并且对于每个开公式 A,只要使 A 中的自由变元指定 U 中个体为值,A 在 U 和 B 中的意义总是相同,就称 U 是 B 的子结构。如果将上述定义中的“每个开公式”改为“每个公式”,则成为初等子结构的定义。例如,$\langle \mathbf{Z};<\rangle$ 是 $\langle \mathbf{Q};<\rangle$ 的子结构,$\langle \mathbf{Q};<\rangle$ 是 $\langle \mathbf{R};<\rangle$ 的初等子结构,其中 $\mathbf{Z}$,$\mathbf{Q}$,$\mathbf{R}$ 分别表示整数集,有理数集、实数集,“$<$”是小于关系。

设 U 和 B 是一阶语言 L 的结构,ψ 是 U 的论域到 B 的论域的双射,如果 ψ 将 U 中每个个体映射到 B 中有同样性质的个体,则称 ψ 是 U 到 B 的同构。如果存在 U 到 B 的同构,则称 U 和 B 同构。同构的结构其逻辑性质完全相同,可将它们看作同一个结构。

称定理相同的理论为等价的理论,如果对于每个闭公式 A,A 和 $\neg A$ 之中恰有一个是理论 T 的定理,则称 T 是完全的理论。如果结构 U 和 B 有完全相同的一阶性质,则对于每个闭公式 A,$U \models A$ 当且仅当 $B \models A$,就称 U 与 B 初等等价。以在结构 U 中有效的闭公式为非逻辑公理的理论称为 U 的理论,记为 $Th(U)$。对于一个有模型的理论 T 来说,以下 3 个条件是等价的:①T 是完全的理论;②T 的任何两个模型初等等价;③T 等价于 $Th(U)$,其中 U 是 T 的一个模型。

如果理论 T 的每个模型 U 的子模型都是 U 的初等子结构,则称 T 为模型完全的理论。如果理论 T 的模型 U 同构于 T 的每个模型的一个子结构,则称 U 为 T 的素模型。一个有素模型的模型完全的理论必是完全的理论。例如,特征为 0 的代数闭域理论是模型完全的理论,并且代数数域是它的一个素模型,因此该理论是完全的理论。

只有一个模型的理论称为范畴的理论。T 是范畴的理论当且仅当 T 是仅有有穷模型的完全理论。范畴性的概念对理论的限制太严了,排除了一切具

有无穷模型的理论，而常用的数学理论许多都是有无穷模型的。因此，在模型论中又引进了 α 范畴的概念。设 α 是一个基数，如果理论 T 只有一个基数为 α 的模型，则称 T 为 α 范畴的理论。关于范畴性的一个著名定理是：如果可数理论 T 对于一个不可数基数 α 是 α 范畴的，则 T 对于任何不可数基数 β 也是 β 范畴的。

由理论的 α 范畴性可以得出它的完全性。设 T 是基数为 α 的没有有穷模型的理论，基数 $\beta \geqslant \alpha$。如果 T 是 β 范畴的理论，则 T 是完全的理论。例如，无最大元和最小元的稠密全序理论是 $\aleph_0$ 范畴的可数理论。它只有无穷模型。因此，它是完全的理论。

模型论与数理逻辑的其他分支有着密切的联系。首先，各种逻辑演算是模型论的基础。此外，在证明论中，有关判定问题的研究广泛使用着模型论方法。在公理集合论中，有关大基数的研究与模型论有密切的联系。另外，布尔值模型被应用于各种独立性问题的研究。公理集合论中的力迫法被移植于模型论中。

模型论与抽象代数、数学分析、数论、拓扑学等数学学科也有联系。模型论中的概念和方法有不少来源于泛代数。模型论的方法被用于解决抽象代数中的某些问题。由鲁滨逊创立的非标准分析则是模型论与数学分析相结合的产物。

模型论的概念和方法也应用于计算机科学和人工智能。实际上，程序设计语言就是一种形式语言，在程序设计语言的语义研究中涉及到许多模型的概念。抽象数据类型的代数语义就是等式逻辑的模型。在非单调逻辑的形式系统及其语义研究中，大量地使用了模型论的概念和方法。

参考文献

1. Chang C C, Keisler H J. Model theory. 3rd ed. Amsterdam: North - Holland, 1990

2. Bridge J. Beginning model theory. Oxford: Clarendon Press, 1977

3. Enderton H B. A mathematical introduction to logic. New York: Academic Perss, 1972

（何自强）

moxing qudong de ruanjian kaifa fangfa

模型驱动的软件开发方法（model-driven software development method, MDSD）　一种以建模（modeling）和模型转换（model transformation）为主要途径的软件开发方法。与传统的软件开发方法相比，模型驱动软件开发方法的特点主要表现在，该方法更加关注为不同的领域知识构造其抽象描述，即领域模型（domain models），基于这些代表领域概念的模型刻画软件系统，并通过自动（半自动）的层层转换完成从设计向实现的过渡，从而最终完成整个系统的开发。

模型驱动开发方法的优势在于，使用更接近于人的理解和认识的模型，尤其是可视化模型，有利于设计人员将注意力集中在和业务逻辑相关的信息上，而不用过早地考虑与平台相关的实现细节。尤其是在面对不同应用领域时，模型驱动方法强调使用方便灵活的领域相关建模语言（domain-specific modeling language, DSML）构造系统的模型，基于领域知识实现领域专家、设计人员、系统工程师以及架构师等不同人员之间的良好沟通。

尽管模型驱动开发方法的研究是在 2000 年左右开始普遍受到关注的，但其中蕴含的思想是软件开发学一直都在研究的。

在软件开发的长期摸索过程中，人们逐渐认识到“提高解决问题的抽象层次”是有效利用抽象手段解决软件开发问题的一个非常具体而实用的途径。Stephen J Mellor 在 21 世纪初曾指出，过去的 50 多年里，人们利用“提高解决问题的抽象层次”处理软件开发的问题已经取得了两个较为显著的进展：①开发出了具有较高抽象层次的程序设计语言；②能够在更高抽象层次上实现软件复用。

有了这样的技术积累，人们开始尝试在更高的抽象层次上开发软件。而正在此时，对象管理组（object management group, OMG）提出了以模型为中心的软件开发框架性标准-模型驱动体系结构（model driven architecture, MDA），受到了来自学术界和工业界的普遍关注。尽管 MDA 提出的直接动因是为了解决异构中间件（middleware）平台的互操作障碍问题，但是由它所倡导的以模型为中心进行软件开发的思想很快得到了广泛支持，迅速成为研究热点。MDA 整合了 OMG 在建模语言、模型存储以及模型交换等方面的一系列标准，形成了一套基于模型技术的软件系统开发方法和标准体系。

随着 MDA 研究热潮的迅速兴起，模型驱动软件开发这个词语逐渐被越来越多的学者使用。此间，和模型相关的不同字眼也不断出现在不同的学术机构和社区中，如 model-driven、model-based、model-related、model-engineering 等。2005 年，模型驱动软件开发领域最重要的年会之一 UML series（International Conference on the Unified Modeling Language）

正式更名为 MoDELS(International Conference on Model Driven Engineering Languages and Systems),这开始引起了人们对模型驱动软件开发领域自身术语使用上的关注。目前,模型驱动软件开发领域较为普遍使用的术语还有模型驱动工程(model-driven engineering,MDE)。

参考文献

1. Mellor S J, Scott K, Uhl A, et al. MDA distilled: Principles of model-driven architecture. Addison Wesley, 2004

2. Object Management Group, Miller J, Mukerji J (eds.) MDA guide version 1.0.1. OMG white paper, 2003

(张天)

moxing xuanze

模型选择(model selection) 指从一个模型空间中找到一个模型,该模型既可以拟合现有的有限训练样本,又对未来样本具有良好的泛化性能。这里的模型是指不同的学习算法或同一算法的不同参数设置(参见**机器学习**)。

“Freedman 悖论”指出当训练样本的数目与变量数目相当时,在缺少先验假设时,不加鉴别地使用模型会导致部分原本与问题不相关的变量在该模型中变得非常重要。

模型选择中存在“偏差-方差两难问题”(Bias-variance dilemma)。简单地说,偏差反映了模型的准确性,方差体现了模型的稳定性。一般情况,如果减小偏差,就会增加模型的复杂度,而如果减少方差,就会减少模型的复杂度。由此导致“偏差-方差两难问题”。

模型选择有两条基本原则,“Occam 剃刀原理”和“没有免费午餐定理”。Occam 剃刀原理提供了一个一般化的模型选择原则:除非必要,“实体”不应该随便增加。换句话说,在很好地拟合样本的前提下,偏爱简单的模型。“没有免费的午餐定理”则指出,不存在某种模型,对所有问题都适用。所以,一定要依据对问题做出的一些假设来选择合适的模型。

常用的模型选择方法可以分为两种。

第一类是经验方法。经验方法不受具体模型的限制。常用的有 *K*-折交叉验证(*K*-fold cross validation)、刀切法(Jackknife)、自助法(Bootstrap)、逐步回归法(Stepwise Regression)。

K-折交叉验证是指样本集被平均分成 *K* 组,模型需要训练 *K* 次,每次留出 *K* 组中的一组作为验证集。当每组中只包含一个样本时,为“留一法”。*K*-折交叉验证方法的缺点是训练次数随着 *K* 增大而增大。当训练算法本身很复杂时,这会带来一些问题。

刀切法是每次从原始样本中留出一个样本,这样共得到等于样本个数的刀切法样本,某个统计量的刀切法估计值定义为对这些样本集的估计值的平均。刀切法适用于统计函数是平滑函数的情况。

自助法对原始样本进行有放回的随机采样,抽取的样本个数同原始的样本个数一样。这个选择过程被独立的重复 *B* 次,由此得到 *B* 个相互独立的自助样本集。某个统计量的自助估计值定义为对独立的 *B* 个自助样本集的估计值的平均。

逐步回归法的主要思路是在待选的全部自变量中按其对因变量的作用大小,显著程度大小或贡献大小,由大到小地逐个引入回归方程,而对那些对因变量作用不显著的变量可能始终不被引入回归方程。另外,已被引入回归方程的变量在引入新变量后也可能失去重要性,而需要从回归方程中剔除出去。

第二类方法是依据不同的模型复杂度理论,通过模型评价准则来选择合适的模型。著名的模型评价准则有赤池信息准则(AIC)、贝叶斯信息准则(BIC)、最小描述长度(MDL)原理、结构风险最小化等。

AIC 基于信息熵概念提出的模型评价准则。AIC 通过参数的个数来度量模型的复杂度,通过对数最大似然来表示模型的准确度,通过二者的简单线性组合判断模型的适合度。AIC 渐进等价于“留一法”。

BIC 是利用贝叶斯定理对指数分布组导出的模型评价准则。BIC 说明用最大似然估计来估计模型参数时,增加参数的数量可能增加似然度,但这会增加模型的复杂度。BIC 不仅考虑模型参数数量,同时考虑了样本数量。相对于 AIC,BIC 更偏爱简单的模型。BIC 渐进等价于 *K*-折交叉验证。

MDL 原理最早提出是为了解决编码问题。对于给定样本,需要保存的样本总描述长度等于这些样本进行压缩编码后的长度加上保存编码模型所需的样本长度。MDL 原理就是选择具有最小 MDL 的模型。MDL 方法等价于 BIC 方法。

结构风险最小化就是在保证经验风险的同时，降低模型的 VC 维，可以使模型在整个样本集上的期望风险得到控制。模型的 VC 维就是该模型能打散的最大样本数目。模型的 VC 维越高模型的复杂度越高。结构风险最小化主要缺点是由其导出的期望误差率界限在实践中很松。

需要指出，在聚类分析中，模型选择通常被称为聚类有效性问题。

参考文献

1. Breiman L. Statistical modeling: the two cultures. Statistical Science. 2001, 16(3): 199-231

2. Hastie T, Tibshirani R, Friedman J. The elements of statistical learning. 2nd ed. Springer, 2009

3. Claeskens G, and Hjort N L. Model selection and model averaging. Cambridge University Press, 2008

（于剑）

N

neibu luyou xieyi

内部路由协议(interior routing protocol)

在一个自治系统(autonomous system, AS)内部用来交换路由信息的协议,也称内部网关协议(interior gateway protocol, IGP)。相对于自治系统之间采用的**外部路由协议**而言,内部路由协议的目的是发现路由的存在,并不操纵这些路由。内部路由协议主要包含两类算法:距离向量路由算法和链路状态路由算法。

距离向量路由算法也叫 Bellman-Ford 算法或 Ford-Fulkerson 算法。使用距离向量路由算法的每个路由器并不知道全局的拓扑信息,而是仅向邻居路由器通告自己已知的到达其他路由器的最短距离(向量)。每个路由器利用这些路由通告更新自己的路由表,并继续向邻居路由器通告新的路由信息。路由信息定期通告。距离向量路由算法存在慢收敛的问题。使用距离向量路由算法的典型协议有路由信息协议(RIP)、内部网关路由协议(IGRP)和增强内部网关路由协议(EIGRP)。

在链路状态路由算法中,每个路由器都拥有完整的网络拓扑信息,这些拓扑信息由链路状态信息组成,保存在每个路由器各自的链路状态数据库中。每个路由器通过发现与它相连链路上的网络前缀来构建链路状态信息,所有路由器通过交换链路状态信息来构建一致的链路状态数据库。根据该数据库,每个路由器独立计算到达每一个目的地的最短路径。使用链路状态路由算法的典型协议包括**开放最短路径优先(OSPF)协议**和中间系统到中间系统(IS-IS)协议。

早期 ARPA 网中使用距离向量路由算法,如**RIP 协议**。RIP 协议适合小型网络系统,在网络规模增大时存在一些问题,还存在无穷计数和慢收敛等不足。所以 1979 年互联网上开始使用链路状态路由算法,例如,OSPF 协议和 IS-IS 协议。

参考文献

Tanenbaum A S. 计算机网络. 4 版. 北京:清华大学出版社, 2004

(徐明伟)

neicun shujuku

内存数据库(main-memory database, MMDB)

使用内存作为常规数据存储设备的数据库。内存数据库有时也被称为 In-Memory database。

与内存数据库相对的磁盘数据库(disk resident database, DRDB)是使用磁盘作为常规数据存储设备,使用内存作为工作数据缓冲区的数据库系统。内存数据库与磁盘数据库的区别如图 1 所示,在内存数据库中,内存作为常规数据存储设备,磁盘是数据的永久存储及后备存储设备;而在磁盘数据库中磁盘是常规数据的存储设备,磁带机是数据的后备存储设备。

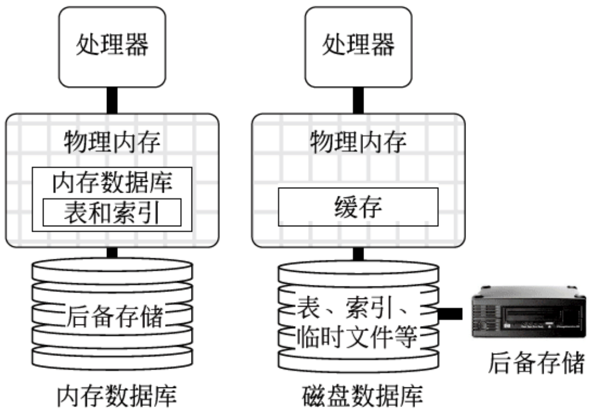


图 1 内存数据库和磁盘数据库的结构

内存数据库中数据常驻内存,消除了磁盘数据库中巨大的 I/O 代价。同时,数据的存储和访问算法以内存访问特性为基础,实现处理器对数据的直接访问,在算法和代码效率上高于以磁盘 I/O 为基础的磁盘数据库。在内存数据库中,使用针对内存特性进行优化的 T 树索引和 hash 索引、面向 cache 优化的索引算法和多种面向连接操作的优化技术,进一步优化了内存数据库的性能。

由于内存是易失性存储介质,因此内存数据库与磁盘数据库相比,在事务的 ACID(atomicity, consistency, isolation, durability)特性上能够满足 ACI 特性,但 D 特性的满足需要借助于特殊的硬件设备、系统设计和实现机制,例如:

（1）日志（transaction logging）　内存数据库需要在事务提交前将日志写到可靠存储设备上，可靠存储设备的访问性能将影响内存数据库的性能。

（2）检查点（check point image）　检查点周期性地将内存数据记录到磁盘上，在发生系统故障并进行恢复时能够还原某一时刻的数据，需要和日志配合使用来保证数据的一致性。

（3）非易失性 RAM（NVRAM）　一种非易失性随机存储设备，包括闪存（flash memory）、PCM（相变存储器）、MRAM（magnetic random access memory）非易失性磁性随机存储器、带有后备电源的 SSD（DRAM 存储）等。

（4）高可用性技术（high availability）　使用数据库复制技术，在发生故障时数据库系统能够自动在多个数据库副本之间进行切换，实现不间断服务，提高数据库的性能和可靠性。

内存数据库一般应用于对实时响应性要求较高的高端应用领域，如电信、金融等领域的核心事务处理（OLTP）。内存数据库既可以作为独立的高性能数据库来处理核心业务，也可以作为磁盘数据库的高速缓存，加速磁盘数据库中"热"数据集的处理性能。在后一种应用模式中，需要对数据库的模式进行优化，划分出"热"数据集和"冷"数据集，由内存数据库和磁盘数据库来分别处理，在两个数据库之间通过数据迁移技术实现底层数据的融合。将内存数据库运行在大内存、多级 cache 和多核硬件环境下，还可以有效解决计算密集型的联机分析处理（OLAP）应用的性能瓶颈。这类分析型内存数据库需要重点解决的问题包括存储模型优化技术、查询处理模型优化技术、轻量压缩技术、cache 优化技术、多核并行查询处理优化技术、cache 分区优化技术等。根据分析型数据的特点，分析型内存数据库一般采用列存储技术和轻量数据压缩技术来提高内存存储效率和访问效率，在连接操作中优化内存带宽和 cache 性能。

随着内存集成度的提高、内存容量的增大和内存成本的降低，高性能内存事务处理和高性能内存分析处理将成为应用的重要目标和实现实时数据处理的关键技术。在内存数据库的研究工作中，既要关注算法优化技术，又要关注硬件发展所带来新的硬件特性，需要在硬件和软件两个层面上进行算法优化技术研究。

参考文献

1. Garcia-Molina H, Salem K. Main memory database system: an overview. IEEE Trans. on Knowledge and Data Engineering, 1992, 4(6): 509-516

2. Manegold S, Boncz P A, Kersten M L. Optimizing main-memory join on modern hardware. IEEE Trans. on Knowledge and Data Eng., 2002: 14(3)

3. Peter A. Boncz. Monet. A next-generation DBMS kernel for query intensive applications. PhD Thesis, Universiteit van Amsterdam, Amsterdam, The Netherlands, May 2002

（王珊　陈红　张延松）

nengli chengshudu moxing

能力成熟度模型（capability maturity model, CMM）　描述和分析软件机构的**软件过程**能力和成熟程度以建立其分级标准和框架的模型。**软件过程能力**是指描述遵循软件过程而得到所期望结果的程度；**软件过程成熟度**是指软件机构中软件过程被明确定义、管理、控制及其实效的程度。利用 CMM 模型，软件机构可以通过评估自己当前的过程成熟程度，来选择相应的改进策略，以达到更高一级的成熟程度；采购机构可以确定与软件机构签订合同的有关风险，并帮助管理正在进行中的合同。

在 20 世纪 30 年代，贝尔实验室物理专家 Walter Shewart 提出了统计质量控制原理。W. Edwards Deming 和 Joseph Juran 在 40～50 年代发展了这些原理，并在实践中得到证明。20 世纪 70 年代末期，ITT 的 Philip Crosby 把这些原理用于构造成熟度框架，并首次提出了质量实践的五个进化阶段。

20 世纪 80 年代中期，IBM 在 Watts S. Humphrey 的指导下，Ron Radice 等人将此成熟度框架首次用于软件过程，并由 Humphrey 于 1986 年将此成熟度框架带到卡内基梅隆大学的软件工程研究所（CMU/SEI）。1987 年 9 月，CMU/SEI 发表了一个简短的软件过程成熟度框架。其后在 Humphrey 的"管理软件过程"一书中进行扩充。在此基础上，1990 年 Jim Withey，Mark Paulk 以及 Cynthia Wise 提出了最早的草案。1991 年 8 月 Mary Beth Chrissis 和 Bill Curtis 帮助 Mark Paulk 校订，提出了 CMM1.1 版。

由于美国联邦政府的大力推行，能力成熟度模型（CMM）得到了广泛应用。据 CMU/SEI 于 2001 年公布的统计，通过 CMM 评估的软件机构共有 964 家。目前 CMM 本身正在向纵深发展，能力成熟度模型族包括：软件能力成熟度模型（SW-CMM），软

件获取能力成熟度模型(SA-CMM),人员能力成熟度模型(People-CMM),系统工程能力成熟度模型(SE-CMM),集成产品开发能力成熟度模型(IPD-CMM),个体软件过程(PSP),群组软件过程(TSP)。另外,国际标准化组织为制订软件过程评价标准,成立了“软件过程改进和能力确定(SPICE)”项目,ISO/IEC JTCI 的SC 7 分技术委员会于 1996 年 9 月正式公布了工作草案,代号为 15504。

能力成熟度模型的组成主要包括过程能力、成熟度级别、关键过程区域、目标、关键实践、关键指示因子和提问单等。CMM 的结构如图 1 所示。

成熟度级别 软件机构向成熟方向迈进的台阶,CMM 有五个成熟度级别。

关键过程区域 一组信息,它指明了一个软件机构为改进其软件过程所应集中关注的区域。实践证明它对改进过程能力最为有效,同时也指明达到一个成熟度级别所必须着手解决的问题和必须满足的要求。CMM 共有 18 个关键过程区域,它分别划归于 1 级除外的其他四个成熟度级别中。

目标 概括和表明了关键过程区域的范围、界限、目的。当该关键过程区域的目标已达到,就可以说,以该关键过程区域为特征的过程能力已被规范化了。

关键实践 对一个关键过程区域的具体化和细节化的描述,为软件机构的过程改进工作提供了具体的指导。当完成关键实践时,将能达到那个关键过程区域的目标。CMM 模型包含多达 316 个关键实践,其描述形式具有相同的特征,即以目标、承诺、能力、活动、监控和验证为标志,分类进行具体、细化的描述。

关键指示因子 过程实践的组成部分,其主要作用是帮助鉴别一个关键过程区域目标是否已经达到。

提问单 列出了有关软件过程的一组“是-否”的提问,它是对每一个关键过程区域的实践的采样,适用于对软件过程能力进行调查研究。

CMM 的能力成熟度级别如图 2 所示。

1 级——初始级 处于 1 级的软件机构,其过程能力是不可预测的,软件项目所涉及的进度、预算、产品功能和产品质量等一般是不可预测的,产品性能只能根据相关人员的个人能力而不是机构的过程能力来预测。

2 级——可重复级 处于 2 级的软件机构的过程能力可以概括为软件项目的策划和跟踪是稳定的。一个有序的管理过程提供了可重复以前成功实践的项目环境。

3 级——已定义级 处于 3 级的软件机构的过程能力可以概括为无论是管理活动还是工程活动都是稳定的。在已建立的产品生产线上,成本、进度、功能和质量都受到控制,而且软件产品的质量具有可追溯性。

4 级——已管理级 处于 4 级的软件机构的过程能力可以概括为软件过程是可度量的。这一级过程能力表明软件机构在定量的范围内预测过程和产品质量趋势,当发生偏离时,能及时采取措施予以纠正,并可预测软件产品是高质量的。

5 级——优化级 处于 5 级的软件机构的过程能力可以概括为软件过程是可优化的,能够持续不断地改进其过程能力,也能借助新技术和新方法来实

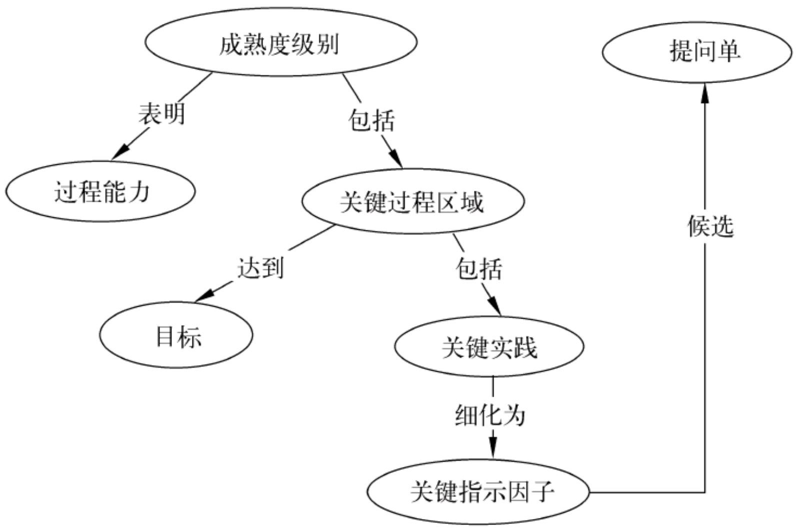


图 1 能力成熟度模型的结构

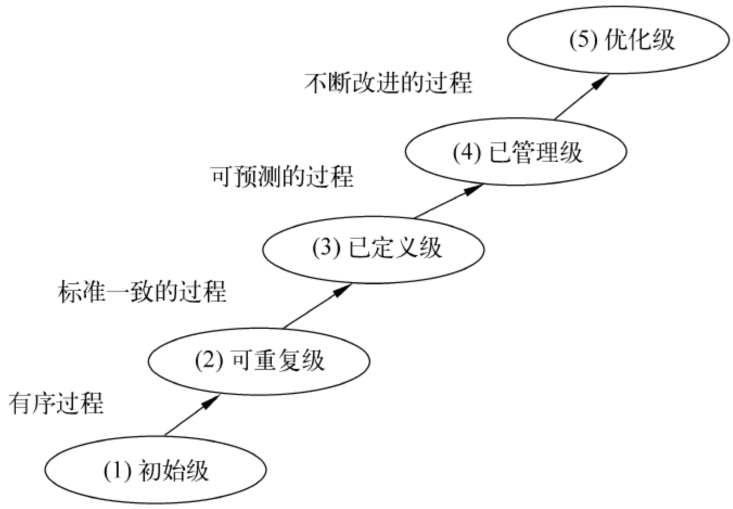


图 2　能力成熟度模型的能力成熟度级别

现未来的过程改进。

能力成熟度模型的广泛采纳和成功推广也推动了软件以外其他相关学科的类似模型的开发。各种模型的推行,出现了许多问题,如过程改善目标和技术的冲突,对培训工作的需求有较大的增长,部分实践人员在应用各种不同的模型来实现特定的需要时产生了混淆等。针对这种情况,美国联邦政府、产业界和 CMU/SEI 于 1998 年启动了"能力成熟度模型集成(CMMI)"项目,于 2000 年第四季度发布了第一个正式的 CMMI,最近的修改稿是 2002 年 6 月 10 日发布的。CMMI 包括了大量的工程改善和过程改善的信息,提供了单一的集成化框架来改善跨学科机构的工程过程,从而提高机构过程改善的质量和有效性。相信 CMMI 将会逐步替代 CMM。2005 年 12 月终止 CMM 的评估,只对软件机构的 CMMI 评估。

参考文献

1. Paulk M C, Curtis B, Chrissis M B, Weber C V. Capability maturity model for software. CMU/SEI-93-TR-24, 1992

2. ISO/IEC TR 15504. Information technology-software process assessment. 1996

3. http://www.sei.cmu.edu/cmmi　（朱三元）

P

paiduilun

排队论(queueing theory) 研究服务系统中排队现象随机规律的一门学科。它的主要目标是研究如何以较少的投资获得最优服务质量的问题。每种服务系统都有三个基本组成:

(1) 输入过程 即顾客(人,信息或作业)到来的规则,通常考虑如下的模型:①定长输入,即顾客等间隔到达;②泊松流输入,即假定在 t 时间内到达 k 个顾客的概率 $P_k(t)=\frac{e^{-\lambda t}(\lambda t)^k}{k!}, k=0,1,\cdots$,$\lambda$ 为单位时间内到达顾客的个数;③其他输入,如一般独立输入、埃尔朗输入、成批到达输入等。

(2) 排队规则 ①损失制,即顾客得不到服务就自动消失;②等待制,其中又分先到先服务、后到先服务、随机服务、优先权服务等方式;③混合制。

(3) 服务机构模型 单个或多个服务台;并联或串联服务台;单个服务或成批服务。服务时间又可分定长分布、指数分布、埃尔朗分布等。

D. G. Kendall 引入一个常用的关于随机服务系统分类的记号 $X/Y/n$,其中 X 为输入分布,Y 为服务分布,n 为服务台数目,并以 M 代表泊松输入或指数的服务分布,D 为定长分布,E_k 为埃尔朗分布,GI 为一般独立输入,G 为一般服务分布。

如果系统已运行了相当长的时间,系统状态不再受初始条件的影响,此时称系统处于稳态,对于 $M/M/n$系统稳态的一个著名公式是埃尔朗公式:顾客到达系统后,由于 n 个服务台被占用而遭受损失的概率为

$$P_n=\frac{\frac{(\lambda/\mu)^n}{n!}}{\sum_{j=1}^{n}\frac{(\lambda/\mu)^j}{j!}}$$

式中,λ 为泊松流强度,μ 为指数服务分布的指数,由此可求得在给定损失率下服务台的最小数目。

另一著名的公式是辛钦公式:在 $M/G/1$ 系统中顾客的平均数

$$Q=\rho+(\rho^2+\lambda^2\sigma^2)/2(1-\rho)$$

式中,$\rho=\lambda/\mu$ 称为服务强度,σ^2 是顾客服务时间的方差。由此可知,在 λ,ρ 不变时,减小 σ^2(即使每个顾客服务时间的差异尽可能地小)可使 Q 下降。

排队论在实时系统、分时系统、计算机网络和存储器分配中都有广泛应用。下面举例说明排队论在操作系统设计中的应用。

(1) 批处理系统和前端处理系统 这里的顾客就是每一个作业或每一条信息,假定它们是泊松输入(参数为 λ),CPU 处理每个作业的时间服从指数分布(参数为 μ)。于是得到一个 $M/M/1$ 模型。以 P_n 表示稳态时系统中有 n 个作业的概率,则有:$P_n=\rho^n(1-\rho)$,其中$\rho=\lambda/\mu<1$。①假定缓冲区容量为 k,则系统因溢出而丢失的概率为 $P_r=\sum_{i\geqslant k+1}P_i=\rho^{k+1}$;②系统中平均等待的作业个数为 $\overline{Q}=\sum_{n=1}^{\infty}nP_n=\rho/(1-\rho)$;③系统中作业的平均等待时间为 $w=\frac{1}{\mu-\lambda}$。

(2) 多用户分时系统模型 假设系统由一台主机 CPU 与 N 个终端构成,输入与服务规律同(1),则可求得在稳态时,系统中没有等待的作业(即全部完成)的概率为

$$P_0=\left(\sum_{i=0}^{\infty}\frac{N!}{(N-i)!}\rho^i\right)^{-1}$$

在交互式分时系统中,响应时间 R 是一个重要的技术指标,它包括用户从键入命令到收到系统响应的时间间隔,可求得平均响应时间 $\overline{R}=\frac{N}{\mu(1-P_0)}-\frac{1}{\lambda}$。假定 $\mu=0.5\text{s}$,$\frac{1}{\lambda}=15\text{s}$,根据上述公式可绘出 $\overline{R}$ 与 N 的关系曲线,并且看出当终端配置超过 20 台后,$\overline{R}$ 将迅速增大。

参考文献

1. 徐光辉. 随机服务系统. 北京:科学出版社,1980

2. 李永锡,等. 计算机操作系统原理. 西安:西北工业大学出版社,1987

(金治明)

paixu suanfa

排序算法(sorting algorithm) 数据处理中将

文件中记录按键码的一定次序要求排列起来的算法。在讨论排序算法时,数据通常是指由若干记录组成的文件,每个记录包含一个或多个数据项,其中能够标志该记录的数据项称为键码。给定一文件的 n 个记录$\{R_1, R_2, \cdots, R_n\}$及其相应的键码集合$\{K_1, K_2, \cdots, K_n\}$,所谓排序就是将记录按键码递增次序排列起来。当待排序的文件能够同时装入计算机的主存中时,则相应的排序称为**内排序**;如果文件大到不能同时全部装入主存中而有一部分必须放在外存上时,则相应的排序称为**外排序**。当待排序的文件中包含有一些相同键码的记录时,如果经过排序后这些相同键码的记录的相对次序仍然保持不变,则相应的排序算法是稳定的,否则为不稳定的。如果排序算法设计成单处理机完成的,则此排序算法称为串行(或顺序)排序算法;如果排序算法设计成多处理机实现的,则称为并行排序算法。度量串行排序算法复杂度的标准是算法的运行时间和所占用的存储空间;度量并行排序算法复杂度的标准是算法的总运行时间和所需的处理器数。排序的应用很广,在科学计算和数据处理中,在数据库和知识库管理系统中,在系统软件和应用软件中以及在高级计算机体系结构中,都会直接或间接地遇到大量的排序问题。排序在计算机科学研究中占有相当的地位,人们已经发现,排序问题的研究方法和思路,算法的设计和分析技巧,对研究计算机诸多领域中其他问题的算法都颇值得借鉴。内排序的方法很多,最常用的有插入排序、选择排序、交换排序(包含快速排序、堆排序)、分配排序和归并排序等。外排序多采用多路归并方法。

插入排序的基本方法是:每次将一个待排序的记录 R_i,按其键码 K_i 的大小插到以前已排序的文件中的适当位置,直到全部插入完为止。

选择排序的基本方法是:每次从待排序的记录中选出其键码最小的记录依次放在已排序的文件中,直到选完为止。

交换排序的基本方法是:两两比较待排序记录的键码,并交换那些不满足顺序要求的键码对,直到全部都满足为止。

分配排序又称为桶排序,它适合于记录有多个特征键码的文件排序。

归并排序的基本思想是:将一些已排序的子文件进行合并而得到一个完整的有序文件。归并时,只要比较各子文件的第一个记录的键码,其最小者就是全局最小者;取出它后,继续比较各子文件的第一个记录的键码,这样就可得到全局的次最小者,如此下去,就可完成排序。

磁盘排序属于外排序。外排序方法与各种外存设备的特征有关。外存设备大体上可分为顺序存取(如磁带)和直接存取(如磁盘)两大类。

磁带排序也是外排序,其过程基本上与磁盘排序过程相同,即先对输入文件的各段进行内排序,生成初始顺串,再把它们写到带上;然后再把这些顺串进行反复地归并,直到剩下一个顺串为止。磁带排序时间主要取决于对带的读写。

并行排序是指利用多台处理机(并行机)进行的排序,其目的主要是为了提高速度。并行排序算法虽然和前述的单处理机上的串行排序算法有不少相似之处,但不能认为它只是串行排序算法的简单推广和扩充。它的最大特点之一就是它和并行计算机的体系结构密切相关,不同体系结构导致不同加速和不同设计风格的并行排序算法。

参考文献

1. 克努特. 计算机程序设计技巧(第三卷:排序和查找). 管纪文,苏运霖,译,陆汝钤,等校. 北京:国防工业出版社,1984

2. 陈国良. 并行算法:排序和选择. 合肥:中国科学技术大学出版社,1990

（陈国良）

Peiteli wanglun

佩特里网论(Petri net theory)　一种以物理学为基础,用计算机科学语言提出的系统模型和理论。简称网论。适合于模拟和分析以资源(物质,数据,信息等)流动为特征的异步并发系统,其最终目标是要用一种兼容物理和计算机两个学科的语言和概念框架来形式描述制约通信进程的所有“自然法则”。

网论起源于 1962 年 C. A. Petri 的博士论文《用自动机通信》。20 世纪 60 年代以研究佩特里网系统(简称网系统)个体的动态行为为主要内容,给出了计算所有可能状态的可覆盖树法,定义了 S-不变量和 T-不变量并给出了计算方法。网系统上的进程由出现网和从出现网到网系统的映射组成,既能正确描述系统中的顺序行为,也能准确表示并发行为。这一阶段的网论又称特殊网论。

20 世纪 70 年代起是佩特里网论发展阶段,至今已有并发论、同步论、赋逻辑论和网拓扑等四个较成熟的分支。它们构成通用网论的基础。

网论尊重物理事件有其自身发生的条件和影响

范围,网系统中没有控制流和中央控制的概念,除非它所描述的物理系统是顺序的或中央控制的。

中央控制需了解系统的全局状态。大系统的全局状态并非总是实时可知的。网论遵循局部确定原则,不依赖全局状态来确定事件之能否发生,因而适合于描述异步并发现象。

网论尊重时间是对变化的度量,认为时间就体现在变化之中,不使用实数作为时间模型。网系统中每个对象的活动轨迹是该对象的全序时间,不同对象交互作用之处则是两个全序时间“同时”之点。网系统中可以读同一个钟表的子系统,可以以此钟表之读数为其局部时间。

网系统以描述资源在系统中的流动为特征。物质流、数据流和信息流均是网系统中的“资源流”。

盛放资源的元素称为库所,改变资源种类和数量的元素称为变迁,连接库所和变迁,代表资源流动方向的是流关系。这三者依次用圆圈、方框和箭头表示时构成网状结构,称为佩特里网,简称网。各类资源的个数用相应圆圈中黑点的个数表示。这种黑点称为托肯。资源在库所中的分布称为网上的标识。网上加上初始标识就是佩特里网系统,简称网系统。标识决定变迁能否发生,变迁的发生则决定标识的变化。规定标识和变迁之间依存关系的法则称为变迁规则。网和标识给出网系统的静态结构,变迁规则给出网系统的动态行为。前者对应于程序系统中的语法,后者对应于语义。

网 由库所集 S,变迁集 T 和流关系 F 构成的三元组 $N=(S,T;F)$ 是个网的条件是:

$S\cup T\neq\varnothing$ 非空

$\wedge S\cap T=\varnothing$ 库所和变迁不同类

$\wedge F\subseteq S\times T\cup T\times S$ ×为笛卡儿积

$\wedge \mathrm{dom}(F)\cup\mathrm{cod}(F)=S\cup T$ 无孤立元素

其中 $\mathrm{dom}(F)=\{x\mid\exists y:\ (x,y)\in F\}$,$\mathrm{cod}(F)=\{y\mid\exists x:\ (x,y)\in F\}$ 分别是 F 的定义域和值域。

网系统 $\Sigma=(S,T;F,K,W,M_0)$ 为网系统的条件是:$(S,T;F)$ 为网,K: $S\to N\cup\{\infty\}$,W: $S\times T\cup T\times S\to N_0$ 及 M_0: $S\to N_0$,其中 N_0 和 N 分别为含 0 和不含 0 的正整数集,且 $W(x,y)=0\Leftrightarrow(x,y)\notin F$。

K 是库所上的容量函数,W 是箭头上的权函数,M_0 是初始标识。

图形表示 图 1 是网系统的图形表示,其中 $S=\{s_0,s_1,s_2,\cdots,s_9\}$,$T=\{t_0,t_1,t_2,\cdots,t_7\}$,$F=\{(s_0,t_1),(s_1,t_1),(t_1,s_3),\cdots,(s_7,t_7),(t_7,s_1)\}$,$M(s_0)=M(s_1)=M(s_2)=1$,而对 $i\geqslant3$,$M(s_i)=0$。

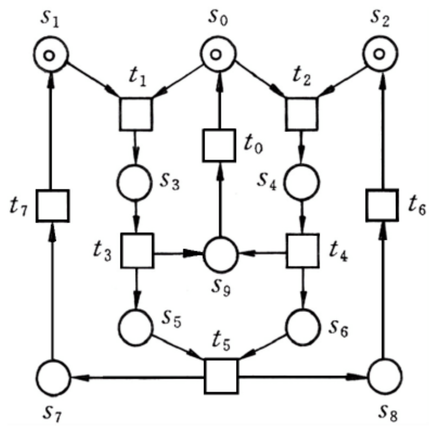


图 1 网系统的图形表示

变迁规则 设 $t\in T$ 为任一变迁,t 的前集 $^{\cdot}t$ 和后集 $t^{\cdot}$ 分别定义为:$^{\cdot}t=\{s\mid(s,t)\in F\}$ 和 $t^{\cdot}=\{s\mid(t,s)\in F\}$。令 M: $S\to N_0$ 为任一标识(即 $\forall s\in S$:$M(s)\leqslant K(s)$),则 t 在 M 有发生权的条件是 $\forall s\in S$: $W(s,t)\leqslant M(s)\leqslant K(s)-W(t,s)$,即其前集库所中有足够的资源,后集库所中有足够的容量。t 在 M 有发生权记作 $M[t>$。此时若 t 发生,则将标识 M 改变为 M 的后继 M',M' 的定义是,对所有 $s\in S$,$M'(s)=M(s)-W(s,t)+W(t,s)$。M' 为 M 的后继,用 $M[t>M'$ 表示。

图 1 中 t_1 和 t_2 各自均能发生,但不能同时发生,因为它们共享的 s_0 类资源只有一个,而它们各需要一个。网论中称 t_1 和 t_2 在 M_0 互相冲突,冲突是网论揭示的基本现象之一。注意图 1 中没有明确给出容量函数 K 和权函数 W。通常这表示容量均为 ∞,权均为 1。

基本现象 包括顺序、冲突、冲撞、并发和混惑等。资源的产生和消耗是顺序关系,共享资源的缺乏引起冲突,容量不够导致冲撞,互不依赖就是并发。混惑是并发和冲突的混合引起的无法判断冲突是否有增减的现象。

并发是提高效率的手段。顺序是被迫的。冲突代表非确定性,是系统环境作决策或行控制之处。冲撞是可以消除的,只要把容量也当作资源直接用库所表示即可。混惑是系统不完整的表现,必须避免。

进程 把系统中变迁的发生和资源的流动如实记录下来,就是一个进程。图 2 给出了图 1 中网系统的一个进程。进程中的网称为出现网。其特点是无有向环路且每个库所至多有一个出的和一个入的箭头。出现网上标出的 $s_0,s_1,\cdots$ 和 $t_0,t_1,\cdots$ 指明各

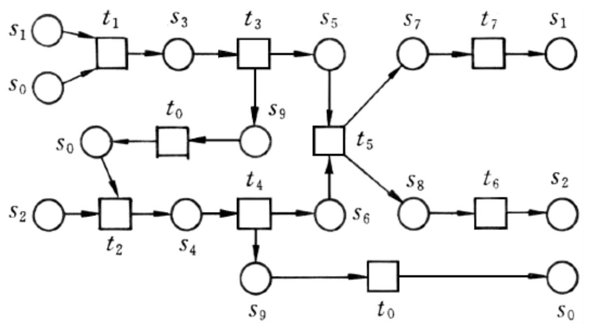


图 2　进程

元素与原系统中库所和变迁的对应。所以网论中的进程是个有序偶$(\mathcal{N},\rho)$，其中$\mathcal{N}$是出现网，ρ是从$\mathcal{N}$到原系统的映射。显然网论的进程准确地表现了原系统中的顺序（如t_1和t_3）和并发（如t_6和t_7）。t_5所起的同步作用也很明显。

网系统是不需加以解释（给出库所和变迁的具体含义）即可分析动态行为的系统模型。

不变量　图 1 中的网系统是个循环系统，因为只要按一定顺序让变迁t_0发生两次，其他变迁各发生一次，它就会恢复到它的初始标识。依次以变迁$t_0,t_1,\cdots,t_7$为序标的列向量$(2,1,1,1,1,1,1,1)^{\mathrm{T}}$称为该系统的$T$-不变量。$T$-不变量也可能只涉及系统的部分变迁。另一方面，无论系统处于从M_0经变迁发生可以到达的什么标识，库所s_0,s_3,s_4和s_9中的托肯总数永远是 1。这就是系统的S-不变量，对应的以$s_0,s_1,\cdots,s_9$为序标的列向量是$(1,0,0,1,1,0,0,0,0,1)^{\mathrm{T}}$。循环系统或循环子系统与$T$-不变量对应，资源全部或部分守恒则对应着$S$-不变量。这些不变量都是与网系统对应的线性方程组的解。

可达标识集　从M_0经变迁发生可以到达的所有标识i集合，可以用来回答系统是否有界，是否安全，是否有死锁及是否有死变迁等问题。网论给出了有效算法，可以构造有限网系统的可覆盖树。所有可达标识都被此树覆盖。

变迁序列　从M_0出发可以依次发生的变迁所成之序列称为变迁序列。所有变迁序列的集合代表着系统的行为，也是构造佩特里网语言的基础。佩特里网语言是将每个变迁映射到一个字符得到的。

高级网系统　不同类的资源可能起着类似的作用，不同的变迁也可能有相似之处。前者如图 1 中的s_1和s_2,s_3及s_4。后者如同一系统中的t_1和t_2,t_3及t_4。将类似的库所和变迁合并，用带记号的托肯（称为个性托肯）代替黑点托肯，就得到高级网系统。这种网系统节点少，便于分析。常用的有谓词/变迁系统和有色网系统。

其他网系统　常见的有随机网系统和时间网系统。

容量函数恒为 1 的网系统是信息系统。这时每个库所代表一位信息。也可将库所看成条件：或成真（有托肯）或不成真，这时对应的变迁称为事件。若像其他网系统一样只研究从初始状态出发的“未来”行为，就得到基本网系统。若还允许逆箭头而“上”，研究它过去的行为，则得到条件/事件系统，简称 C/E 系统。C/E 系统是通用网论的基础。通常用B和E代替S和T分别表示条件集和事件集，而且把 C/E 系统中正向和反向可以到达的标识称为情态。

同步距离　任取$E_1,E_2\subseteq E$，用E_1作前集，E_2作后集，可以构造出一个S-类元素。这个元素若不属于B，即不是系统中给出的条件元素，就可以用来观察E_1事件和E_2事件的动态关系。办法是把它当作容量为无穷的库所，而且假定其中有足够多的托肯。当E_1中事件正向发生一次或E_2中事件反向发生一次，它得到一个托肯；若E_1中事件反向发生一次或E_2中事件正向发生一次，它失去一个托肯。其中托肯总数最大值和最小值的差就是E_1和E_2之间的同步距离，记作$\sigma(E_1,E_2)$。若最大值不存在，$\sigma(E_1,E_2)=\infty$，$\sigma(E_1,E_2)$满足距离公理。已证明，顺序事件的同步距离为 1，并发或冲突事件的同步距离则不小于 2。同步距离是同步论的核心概念。

赋逻辑　若$B_1,B_2\subseteq B$为不相交的两个条件子集，则分别以B_1,B_2为前集和后集可以构造出一个T-类元素。若这个T-类元素在任何情态下均不能发生，就说它是死的。设$B_1=\{a_1,a_2,\cdots,a_n\}$，$B_2=\{b_1,b_2,\cdots,b_m\}$，则这个死的$T$-类元素对应着命题$a_1\wedge a_2\wedge\cdots\wedge a_n\rightarrow b_1\vee b_2\vee\cdots\vee b_m$在所有情态成真。这样，死的$T$-类元素就可以用来对 C/E 系统的情态进行命题演算。已证明关于情态的任何有效命题均可用一个或几个这种死元素表示。死元素上的扩张规则和消解规则及反证法（当$n=0=m$时，死元素代表矛盾）可以完成命题演算及命题证明。这些都可以代数化，用矩阵的初等变换来完成。

C/E 系统和模态逻辑和时态逻辑同样密切相关。

并发论　网论提出了 22 条并发公理，以刻画并

发现象的根本特性。例如并发关系的不传递性及其传递核和传递闭包;并发的非平凡性,自返性,对称性,不可化简性,相干性,K-稠密性,N-稠密性及并发的连续性。

网拓扑 拓扑学中的开集公理来源于实数轴。由此导出的连续性隐含着全序、稠密及实数势,而且每个点都成闭集。实际应用中往往要偏序集,有限或可数无限集上的连续性和稠密性。将开集公理中有限个开集之交仍为开集放松为任意多个开集之交仍为开集,就得到网拓扑。这是沟通连续模型和离散模型的途径。

网系统允许层次模拟,从最细节的出现结构到最抽象的系统,都以网结构为基础。不同层次之间可以由在网拓扑之下的连续映射沟通。这就使网论应用有了理论基础。佩特里网论已经或正在成功地应用于网络通信、软件工程、柔性制造、系统控制及CAD 等领域。工业上的国际和国家标准《控制系统功能图表的绘制》即是以佩特里网为基础制定的。目前已有构造和分析网系统的计算机辅助工具上市。

参考文献

1. Brauer W. Net theory and applications. LNCS Vol. 84, Springer-Verlag, 1979

2. Brauer W, Reisig W, Rozenberg G. Petri nets: applications and relationships to other models of concurrency. LNCS Vol. 254, 255, Springer-Verlag, 1986

3. 袁崇义. Petri 网. 南京:东南大学出版社, 1989

(袁崇义)

penquan moxing

喷泉模型(fountain model) 一种体现软件创建所固有的迭代和无"间隙"特征的软件开发模型。如图 1 所示。

这一模型刻画了软件开发活动需要多次重复。例如,在编码之前(设计之后),再进行分析和设计,期间,添加有关功能,使系统得以演化。同时,该模型还表明活动之间没有明显"间隙",例如分析活动和设计活动之间没有明显的界线。

喷泉模型主要用于描述面向对象开发过程。由于对象概念的引入,表达分析、设计、实现等活动只用对象和关系,因而,可以容易实现活动的重复以及活动之间无明显"间隙"。并且,使其开发自然地包括复用。 (王立福)

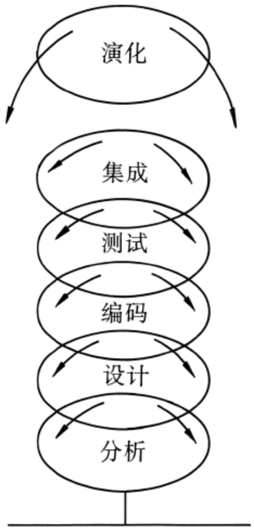


图 1 喷泉模型

pichuli

批处理(batch processing) 把多个作业积累成批,处理过程中用户不能再行干预的处理方式。

采用批处理方式处理作业时,用户把要处理的作业,包括程序、数据及作业说明书一起交操作员或通过终端输入到计算机,然后,由系统按作业说明书来控制执行。

提供批处理功能的操作系统称**批处理操作系统**,它根据预定的策略按某种组合和次序选择待处理的作业去执行。例如,选择计算型和输入输出型用户作业搭配运行,可使系统各类资源均衡使用,提高系统效率。

参考文献

孙钟秀,等. 操作系统教程. 北京:高等教育出版社,1989

(郑宇华)

pianshang wangluo

片上网络(network-on-chip, NoC) 一种连接**片上系统**各个模块的通信系统。片上网络把网络设计的理论和方法应用到了片上通信机制中,从而提供了比传统总线和交叉开关更好的性能。相对于其他互联机制,片上网络改善了片上系统的扩展性和能效。

片上网络是一种在众核处理器或片上系统等复杂集成电路设计中实现互联机制的新兴技术。典型的片上网络结构包括环、网格(Mesh)、环网(Torus)等。在一个片上网络系统中,处理器核、存储模块和其他模块(例如知识产权(IP)模块)通过一个"公共

的传输子系统”来交换数据。片上网络由多个点对点的数据链路组成，这些数据链路由交换引擎（片上路由器）连接起来。通信的消息可以从源模块出发，跨越多条数据链路然后到达目的模块，其中，路由决策是在交换引擎中完成的。片上网络在复用的链路上采用比特/分组交换的传输方式。分组交换是片上网络的常见特点，电路交换的技术也被应用于某些片上网络的设计。在片上网络中，多条通信链路可以同时传输不同的分组消息包。因此，相比于传统的通信结构（例如专用的点对点总线、共享总线、分段桥接总线、交叉开关等），片上网络提供了更好的性能和扩展性。

传统的集成电路中，采用互联线机制通过点对点的连接实现通信。随着集成电路设计与制造工艺的进步，片上系统集成度越来越高，传统通信机制不再适用。在超深亚微米工艺下，过多信号线增大芯片面积，增加动态功耗，增大线间串扰概率，阻碍性能提升。

片上网络系统由于结构规整大大降低了通信结构的设计复杂度。从系统设计者角度来看，在多核处理器的研制中，片上网络是一个很好的架构选择。片上网络系统可以方便地隔离计算和通信，支持模块化设计和IP重用（通过标准的接口界面），提高工程效率。片上网络中所有的链路可以同时传输不同的数据包，具有很高的并行性。片上网络相对于传统的互联结构能够提供更好的性能（例如吞吐量）和扩展性。

片上网络方法学可以借助成熟的**计算机网络**领域的概念和技巧，但不能盲目搬用传统计算机网络和对称多处理器的设计方法。片上网络的交换引擎需面积开销小、能效高、且延迟小。片上网络系统的路由算法应简单易行，路由缓冲应尽可能小，网络拓扑应基于目标应用程序优化设计。保证片上网络的服务质量也是此领域研究的一个重点，它体现了目标应用在吞吐量、实时延迟和截止时间等各方面的需求满足情况以及并发多用户在单芯片上对片上资源的共享需求情况等。

参考文献

Dally W J, Towles B P. Principles and practices of interconnection networks. San Francisco: Morgan Kaufmann, 2004

（范东睿）

pianshang xitong

片上系统（system on a chip，SoC） 具有完整系统功能并有专用目标的一种复杂芯片。至少包括处理器、局部存储器、接口和专用处理部件以及内嵌的基本软件模块或可载入的用户软件等。它源于1998年前后，是超大规模集成电路之后新发展阶段的标志，支持它的技术是深亚微米互补金属-氧化物-半导体（CMOS）工艺和可将上亿量级（10^8）的器件集成到单一芯片的工艺。

SoC的应用范围广泛，内涵多种多样。其中文名称尚未统一，常见的有单芯片系统、系统级芯片、系统集成芯片等。

一种集成电路芯片如果具备了下述特征，可称为SoC。这些特征是：①具有实现复杂系统功能的VLSI；②采用深亚微米工艺技术；③使用一个或数个嵌入式CPU、MPU或DSP；④具有从外部对芯片进行编程的功能；⑤配套有嵌入式软件和操作系统或实时操作系统；⑥主要采用第三方的IP（知识产权）核进行设计。为了缩短进入市场时间，SoC采用已经被验证过的IP核进行设计，因此，可重用的成功设计越多，其设计过程会越短。

SoC的内涵随技术的进步而改变，由于它比一般的集成电路具有更高的集成度，因而更适合于专用。SoC在开始时的应用领域主要是个人通信类产品和信息家电产品。随着技术的进步，无线SoC得到了广泛应用，除了SoC的一般模块外，还将射频前端模块、模数或数模转换模块、电源提供和功耗管理模块等都集成在同一个SoC中。

SoC设计的关键技术主要包括总线架构技术、IP核可复用技术、软硬件协同设计技术、SoC验证技术、可测性设计技术、低功耗设计技术、超深亚微米电路实现技术等。例如，对SoC一类的复杂芯片，测试是一个难题。因此，在设计的各个阶段都要考虑可测试性问题，这样可测试性设计技术（DFT）便得以发展。SoC的设计过程包括体系结构和算法设计、IP选择、DFT设计、系统集成、验证及物理综合等。

为了进一步缩短进入市场的时间，降低SoC的开发风险、开发成本和小批量生产成本，一些半导体生产厂商推出了多种可编程片上系统产品，即可编程的片上系统（system on a programmable chip，SoPC）产品。SoPC在芯片中预先定制了一些CPU的硬核、静态随机存储器（SRAM）、定点乘法器、双端口存储器和接口电路，并提供几十万至几百万可编程逻辑门。此外，还可以提供各类IP核（参见**知识产权核**）。这类芯片兼具定制逻辑与可编程逻辑的优

点,提高了 SoPC 的性能,缩短了产品的开发风险和开发周期,是 SoC 的主要发展趋势之一。

信息技术正在逐步改变着人们的日常生活。在这个改变过程中,SoC 做出了重大贡献,也取得了飞速发展。

参考文献

郭炜,等. SoC 设计方法与实现. 北京:电子工业出版社,2007
(时万春 韩承德)

pianweifen fangcheng shuzhi jiefa

偏微分方程数值解法(numerical solution of partial differential equations) 根据偏微分方程特点,研究在计算机上如何采用有效的数值方法求解偏微分方程的方法和理论。用离散的网格覆盖求解区域,按某种方式将连续形式的偏微分方程离散成以网格点(结点)上的函数值为未知量的代数方程组,求解方程组得到结点上未知函数的值。这是大型数字计算机在科学技术领域的重要应用,涉及科学与工程计算的许多问题,范围相当广泛。

偏微分方程是联系自变量与未知函数及其偏导数的等式。一个重要的类型是两个变元的二阶线性偏微分方程,其一般形式是

$$\mathrm{L}u = Au_{xx} + 2Bu_{xy} + Cu_{yy} + Du_x + Eu_y + Fu + G = 0 \qquad (1)$$

其中 L 是微分算子,u 是未知函数,A,B,C,D,E,F 和 G 依赖于 x 和 y,u_x 表示 u 对 x 的偏导数。方程(1)在点(x,y)是椭圆型、双曲型或抛物型,按判别式 $B^2 - AC$ 在该点为负、为正或为零而定。典型的例子有

$$\Delta u = f(x,y) \quad (\text{椭圆型}) \qquad (2)$$

$$\frac{\partial^2 u}{\partial t^2} = a^2 \frac{\partial^2 u}{\partial x^2} \quad (\text{双曲型}) \qquad (3)$$

$$\frac{\partial u}{\partial t} = a^2 \frac{\partial^2 u}{\partial x^2} \quad (\text{抛物型}) \qquad (4)$$

其中 $\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ 称为拉普拉斯算子,$f(x,y)$ 是已知函数,a 为常数。以上三个方程也依次分别称为泊松方程($f(x,y) = 0$ 时称为拉普拉斯方程)、波动方程和热传导方程。

偏微分方程的两类重要定解问题是边值问题和初值问题,以方程(2)和方程(4)为例有下面的定义。

求 $u(x,y)$ 在有界区域 Ω 内满足方程(2),在 Ω 的边界上满足 $u = \varphi(x,y)$(φ 是已知函数),称这样的定解问题为狄利克雷问题或第一边值问题;若在边界上 $\partial u/\partial n = \varphi$($n$ 为边界的外法线方向),称这样的边值问题为诺依曼问题或第二边值问题。

求 $u(t,x)$ 在区域 $\{|x| < \infty, t > 0\}$ 内满足方程(4),$t = 0$ 时 $u(0,x) = u_0(x)$(u_0 是已知函数),称这样的定解问题为初值问题或柯西问题。

如果一个定解问题的解存在、唯一且连续地依赖于定解条件,则此定解问题称为适定的。定解问题只有在很特殊的情况下才能用解析的方法求解,一般情况下采用数值方法,如差分法、**有限元方法**等求解。

差 分 法

差分法是把定解区域剖分为网格,在网格结点上用差商代替微商,将微分方程化为包含有限个未知数的差分方程组。差分法直观、简单易行,能普遍地用于各种类型的微分方程和任意形状的区域。但因为它包含巨大的运算量,只有在电子计算机问世之后才得到广泛的应用和发展。

边值问题差分法 求解椭圆型方程边值问题的一类数值方法。作为例子考虑泊松方程的狄利克雷问题:

$$\left.\begin{aligned} \Delta u &= f, \text{在 } \Omega \text{ 内} \\ u|_{\partial\Omega} &= \varphi, \text{在 } \partial\Omega \text{ 上} \end{aligned}\right\} \qquad (5)$$

其中 $\partial\Omega$ 表示区域 Ω 的边界。

区域 Ω 的网格剖分,一种最简单又常用的方法是以平行于坐标轴的两族直线为网格线,作正方形网格剖分得网格 Ω_h(图 1)。结点坐标为 $x_i = ih$,$y_j = jh$,h 称为步长(两相邻平行线间的距离),i,j 为整数。记 $u_{ij} = u(x_i,y_j)$,当(x_i,y_j)是内点时,用差商 $(u_{i+1\,j} + u_{i-1\,j} - 2u_{ij})/h^2$ 代替 $\partial^2 u/\partial x^2$,用差商 $(u_{i\,j+1} + u_{i\,j-1} - 2u_{ij})/h^2$ 代替 $\partial^2 u/\partial y^2$,得到微分方

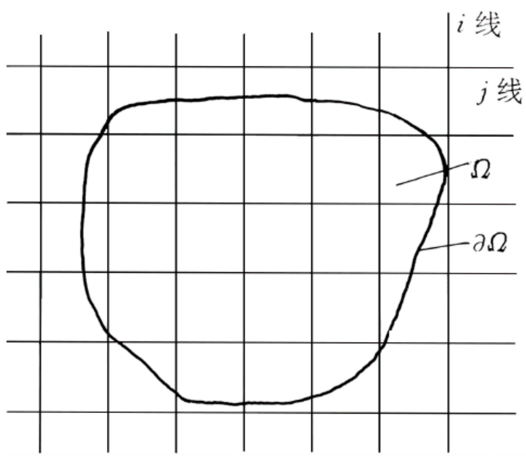


图 1

程的差分方程(差分格式)

$$(u_{i+1\,j} + u_{i\,j+1} + u_{i-1\,j} + u_{i\,j-1} - 4u_{ij})/h^2 = f_{ij} \tag{6}$$

格式(6)常称为五点差分格式,可简记为:$\Delta_h u_h = f_h$,其中 Δ_h 是对应于微分算子 Δ 的差分算子,u_h 和 f_h 是定义在 Ω_h 上的网函数:$u_h(ih,jh) = u(ih,jh)$,$f_h(ih,jh) = f(ih,jh)$。

对于靠近边界的结点,如果结点正好落在边界上,则取相应的边界值。一般情况下,用偏心差商列出不等距差分方程,也可用其他方法(如插值法)作边界处理。

对于一般的微分方程 $\mathrm{L}u = f$,类似的方法用差分算子 L_h 代替微分算子 L 逼近微分方程,得到差分方程 $\mathrm{L}_h u_h = f_h$。当 u 充分光滑时,称 $R_h(u) = [\mathrm{L}_h u]_{ij} - [\mathrm{L}u]_{ij}$ 为差分方程的截断误差,其中 h 的最低幂次称为截断误差的阶。当网格步长趋于零时,差分格式的截断误差也趋于零,则称差分格式是微分方程的相容逼近。相容性说明网格步长越小,差分方程与微分方程越接近,截断误差的阶反映了逼近的精度。

差分方程 $\mathrm{L}_h u_h = f_h$(含边界条件)实际上已是线性或非线性方程组,常用的求解方法是直接法和迭代法。针对差分方程的特点,现已开发了一些求解偏微分方程的软件包,如 ELLPACK,它由输入、划分网格、离散化、方程求解及输出等模块组成。用迭代法求解椭圆差分方程组时,可应用 ITPACK 软件包中的有关模块。用直接法时可用 LINPACK 或稀疏矩阵软件包 Yale 中有关模块。

初值问题差分法　它是求解双曲型方程和抛物型方程这类含时间 t 的偏微分方程的初值问题或初边值混合问题的一类数值方法。以一阶双曲型方程的初值问题为例:

$$\left.\begin{aligned} &\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = 0, \quad |x| < \infty, t > 0 \\ &u(0,x) = u_0(x), \quad |x| < \infty \end{aligned}\right\} \tag{7}$$

取时间步长 τ 和空间步长 h,作 xt 平面的网格剖分(图2)。记 $u_j^n = u(n\tau, jh)$,用差商 $(u_j^{n+1} - u_j^n)/\tau$ 代替 $\partial u/\partial t$,若用一阶中心差商 $(u_{j+1}^n - u_{j-1}^n)/(2h)$ 代替 $\partial u/\partial x$,则得式(7)的一个差分格式

$$u_j^{n+1} = u_j^n - \frac{1}{2}r(u_{j+1}^n - u_{j-1}^n) \tag{8}$$

当 $a > 0$ 时,可用向后差商 $(u_j^n - u_{j-1}^n)/h$ 代替 $\partial u/\partial x$,得格式

$$u_j^{n+1} = u_j^n - r(u_j^n - u_{j-1}^n) \tag{9}$$

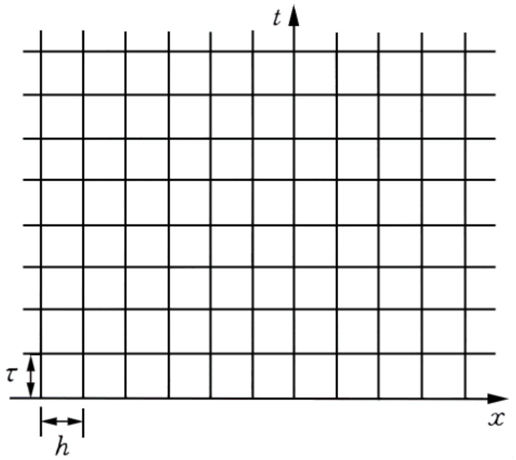


图　2

如果 $a < 0$,则用向前差商 $(u_{j+1}^n - u_j^n)/h$ 代替 $\partial u/\partial x$,得格式

$$u_j^{n+1} = u_j^n - r(u_{j+1}^n - u_j^n) \tag{10}$$

其中 $r = a\tau/h$。

格式(8)的截断误差是 $O(h^2 + \tau)$,格式(9)和格式(10)的截断误差是 $O(h + \tau)$,它们都满足相容性。根据差分格式以及初值条件 $u_j^0 = u_0(jh)$,可依次求出 $u_j^1, u_j^2, \cdots$。

差分格式的稳定性　对于偏微分方程的初值问题,差分格式的稳定性对于实际数值计算至关重要。若稳定性的研究主要针对初始条件,则称为关于初值的稳定性。它要求初值 u_j^0 小的改变引起 u_j^n 的改变也小。对于常系数线性偏微分方程的差分格式的稳定性,诺依曼运用傅里叶分析作了系统的研究。把差分方程的解表示成谐波的叠加。考虑其中一个谐波

$$u_j^n = G^n(\sigma,\tau)\mathrm{e}^{\mathrm{i}j\sigma h} \tag{11}$$

的增长情况。其中 σ 为实数,i 为虚数单位,$G(\sigma,\tau)$ 称为增长因子。若对一切谐波(11)的振幅一致有界,即存在常数 $M > 0$,对一切适合 $0 \leqslant n\tau \leqslant T$($T$ 为某一取定的常数)的 n 和充分小的 τ 都有 $|G^n| \leqslant M$,则此差分格式是稳定的。

显式格式与隐式格式　考虑抛物型方程的初值问题

$$\left.\begin{aligned} &\frac{\partial u}{\partial t} = a^2\frac{\partial^2 u}{\partial x^2}, \quad |x| < \infty, t > 0 \\ &u(0,x) = u_0(x), \quad |x| < \infty \end{aligned}\right\} \tag{12}$$

记 $\delta^2 u_j = u_{j+1} - 2u_j + u_{j-1}$,用差商 $(1/h^2)\delta^2 u_j^n$ 代替 $\partial^2 u/\partial x^2$,$\partial u/\partial t$ 仍用 $(u_j^{n+1} - u_j^n)/\tau$ 代替,可得差分格式

$$u_j^{n+1} - u_j^n = r\delta^2 u_j^n \tag{13}$$

这里 $r = a^2\tau/h^2$。若初值是以 1 为周期的函数,且假设 $u_0(0) = u_0(1) = 0$,设 $N = 1/h$,则还应加上边界条件:$u_0^{n+1} = u_N^{n+1} = 0$。

若用差商 $(1/h^2)\delta^2 u_j^{n+1}$ 代替 $\partial^2 u/\partial x^2$ 时,则得差分格式

$$\left.\begin{aligned} &u_j^{n+1} - u_j^n = r\delta^2 u_j^{n+1}, \quad j = 1,2,\cdots,N-1 \\ &u_0^{n+1} = u_N^{n+1} = 0 \end{aligned}\right\} \tag{14}$$

比较这两个格式,前者可以由 n 层结点上的 u_j^n 直接计算 u_j^{n+1},这称为显式格式。后者是含未知数 $u_j^{n+1}(j = 1,2,\cdots,N-1)$ 的方程组,不能直接计算 u_j^{n+1},必须解方程组,故称为隐式格式。

拉克斯等价定理 对于线性偏微分方程组的适定的初值问题,一个与之相容的线性差分格式收敛的充分必要条件是这个格式是稳定的。

多重网格法

该法是求解差分方程组(网格方程)的一类数值方法。

多重网格法将求解区域用不同步长进行剖分,得到多个层次的由粗到细逐步加密的网格,把细层网格上求解问题转化为较粗层(直到最粗层)网格上求解。

多重网格迭代是以二重网格迭代为例,考虑边值问题(5)。用两种步长 h 和 $H(h < H)$ 剖分得网格 Ω_h 和 Ω_H,并有差分算子 L_h 和 L_H,Ω_h 上的离散方程是

$$L_h u_h = f_h \tag{15}$$

设 u_h^j 是式(15)的一近似解,用 $e_h^j = u_h - u_h^j$ 表示 u_h^j 与离散方程精确解 u_h 的误差,$d_h^j = f_h - L_h u_h^j$ 是 u_h^j 的残量,从而有误差方程

$$L_h e_h^j = d_h^j \tag{16}$$

由式(16)的解和近似解 u_h^j 可得式(15)的解:$u_h = u_h^j + e_h^j$。

线性转换算子 I_h^H 与 I_H^h 分别称为限制算子和延拓算子,$I_h^H: G(\Omega_h) \to G(\Omega_H)$,$I_H^h: G(\Omega_H) \to G(\Omega_h)$。其中 $G(\Omega_h)$ 表示 Ω_h 上的网函数空间。又设给定了解 Ω_h 上的离散方程(15)的一个松弛法:$\bar{u}_h^j = S(u_h^j, L_h, f_h)$,并用 S^k 表示执行 k 次 S 松弛,则有二重迭代过程如下:

(1) 对 u_h^j 用松弛法 S 作 k_1 次松弛得 $\bar{u}_h^j = S^{k_1}(u_h^j, L_h, f_h)$。

(2) 粗网格校正得 $\bar{\bar{u}}_h^j$:① 计算残量 $\bar{d}_h^j = f_h - L_h\bar{u}_h^j$;② 限制残量 $\bar{d}_H^j = I_h^H \bar{d}_h^j$;③ 在 Ω_H 上解残量方程 $L_H \bar{e}_H^j = \bar{d}_H^j$ 得 $\bar{e}_H^j$;④ 延拓校正得 $\bar{e}_h^j = I_H^h \bar{e}_H^j$;⑤ 计算校正近似值 $\bar{\bar{u}}_h^j = \bar{u}_h^j + \bar{e}_h^j$。

(3) 对 $\bar{\bar{u}}_h^j$ 用松弛法 S 作 k_2 次松弛得 $u_h^{j+1} = S^{k_2}(\bar{\bar{u}}_h^j, L_h, f_h)$。

以上过程反复进行直到某个 u_h^{j+1} 满足精度要求为止。(1)和(3)中的 k_1 和 k_2 的选取依具体问题而定,尚无统一的选取原则。由上述迭代过程可见,多重网格法是一个细网格上松弛和粗网格上校正来回交替迭代的过程,其优点是计算量较小而收敛速度快。

区域分裂法

并行求解偏微分方程特别是椭圆型方程的一类数值计算方法。

区域分裂法将定解问题的求解区域分成 m 个子区域 Ω_i:$\Omega = \bigcup_{i=1}^{m} \Omega_i$。相应地将原问题分解成 m 个较小的子问题,每个子问题独立求解,同时还须建立子问题之间的某些相互联系与通信,以保证它们的总体效果收敛到原问题的解。这种思想早在 1870 年就由德国数学家 H. A. Schwarz 提出,并用于解非凸平面区域上拉普拉斯方程的狄利克雷问题。1890 年 E. Picard 用它解一类非线性椭圆型方程,称之为施瓦茨交替法。但直到 20 世纪 70 年代末期,它才随着并行计算机的问世而发展成为解偏微分方程的一类并行数值方法。此后产生了子区域重叠和不重叠的区域分裂法,对称区域分裂法等多种类型,并由此产生了空间分裂法等。

施瓦茨交替法 考虑平面拉普拉斯方程狄利克雷问题

$$\left.\begin{aligned} &\Delta u = 0, \quad 在\ \Omega\ 内 \\ &u|_{\partial\Omega} = \varphi \end{aligned}\right\} \tag{17}$$

其中边界 $\partial\Omega$ 由不相切的两部分 $\overset{\frown}{ACB}$ 和 $\overset{\frown}{ADB}$ 组成(图 3)。将 Ω 分裂成两个子区域 Ω_1 和 $\Omega_2(\Omega_1 \cap \Omega_2 \neq \varnothing)$,它们的边界分别是 $\overset{\frown}{ACBFA}$ 和 $\overset{\frown}{AEBDA}$,其中 $\overset{\frown}{AEB}$ 和 $\overset{\frown}{AFB}$ 称为拟边界。于是原问题分解成两个子问题:

$$\left.\begin{aligned} &\Delta u = 0, \quad 在\ \Omega_1\ 内 \\ &u|_{\overset{\frown}{ACB}} = \varphi, \quad u|_{\overset{\frown}{AFB}} = g_1 \end{aligned}\right\} \tag{18}$$

$$\left.\begin{aligned} &\Delta u = 0, \quad 在\ \Omega_2\ 内 \\ &u|_{\overset{\frown}{ADB}} = \varphi \\ &u|_{\overset{\frown}{AEB}} = g_2 \end{aligned}\right\} \tag{19}$$

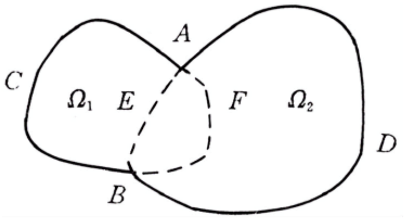


图 3

其中 g_1 和 g_2 分别是 $\widehat{AFB}$ 和 $\widehat{AEB}$ 上的连续函数，且 $g_1(A)=\varphi(A)=g_2(A)$，$g_1(B)=\varphi(B)=g_2(B)$，它们由交替迭代过程确定并随迭代步变化。

解问题(17)的施瓦茨交替法如下：

(1) 给问题一个初始猜测 u^0；

(2) 令 $g_1|_{\widehat{AFB}}=u^0|_{\widehat{AFB}}$，求解问题(18)得 $\bar{u}^1$，延拓 $\bar{u}^1$ 到 Ω 得

$$u^1=\begin{cases}\bar{u}^1, \text{在 } \Omega_1 \text{ 内} \\ u^0, \text{在 } \Omega\backslash\Omega_1 \text{ 内}\end{cases};$$

(3) 令 $g_2|_{\widehat{AEB}}=u^1|_{\widehat{AEB}}$，解问题(20)得 $\bar{u}^2$，同法延拓得 u^2。

重复执行(2)和(3)两步，即交替地解两个子问题，得到函数序列 $u^0,u^1,\cdots,u^{2n+1},u^{2n+2},\cdots$。此序列的极限即为问题(17)的解，这由下面的定理所保证。

施瓦茨定理　若问题(17)有唯一解且它在 Ω 内二次连续可微，则序列 $\{u^i\}$ 在 Ω 的任一内子区域上一致收敛于此解。

在交替过程中还可以引进松弛因子 ω，若记(2)和(3)的解为 $\hat{u}^{2n+1}$ 和 $\hat{u}^{2n+2}$，并分别加入下列松弛过程

$$u^{2n+1}=u^{2n}+\omega(\hat{u}^{2n+1}-u^{2n}) \qquad (20)$$

$$u^{2n+2}=u^{2n+1}+\omega(\hat{u}^{2n+2}-u^{2n+1}) \qquad (21)$$

其中 $0<\omega<2$。这样做成的序列 $\{u^i\}$ 也收敛到问题(17)的解，且若 ω 选得合适，则它较原序列收敛得更快。这就是带松弛因子的施瓦茨交替法。

混乱松弛与并行算法　松弛步(2)和(3)是有先后顺序的，它的直接推广是将 Ω 分裂为 m 个子区域的系统松弛法。它按区域 Ω_j 的编号顺序循环地松弛得 $u^{mt+j}(t=0,1,2,\cdots;j=1,2,\cdots,m)$，即第 $mt+j$ 次松弛总是在第 j 个子区域上进行。混乱松弛则是松弛次序允许千变万化。记 $i(j)$ 表示第 i 次松弛是在第 j 个子区域上进行，则混乱松弛法可描述为：给一初始猜测 u^0，在 Ω_k 上松弛得 $u^{1(k)}$，在 Ω_l 上松弛得 $u^{2(l)},\cdots$，照此下去得到序列 $\{u^{i(j)}\}(i=1,2,\cdots;k,l,j=1,2,\cdots,m)$。这样做成的序列并不能保证收敛，因为不加限制的标号序列 $\{i(j)\}$ 可能导致有的子区域无限次被松弛，而有的子区域只松弛有限次。为此将序列 $\{i(j)\}$ 按子区域的编号分成 m 个子序列：$\{i_1(1)\}$，$\{i_2(2)\},\cdots$，$\{i_j(j)\},\cdots$，$\{i_m(m)\}$。$\{i_j(j)\}$ 是对第 j 个子区域松弛的那些序号的序列，并规定当 $i\to\infty$ 时，有 $i_j(j)\to\infty(j=1,2,\cdots,m)$。有了这样的规定，则 $\{u^{i(j)}\}$ 收敛到问题(17)的解（在施瓦茨定理的条件下）。

混乱松弛的计算机实现便是并行算法。设有 m 台处理机，每台处理机负责一个子区域上的松弛过程。这些过程在运行时除了拟边界上的近似解的最新信息相互交换之外，各自独立并行执行，彼此不需要等待，故是一类异步并行算法。当然为了保证必要的精度要求与逻辑的正确性，还应当有某些作为判断依据的控制信息参与过程之间的通信。

参考文献

1. 李荣华，冯果忱. 微分方程数值解法. 北京：高等教育出版社，1980

2. 哈克布思 W. 多重网格法. 北京：科学出版社，1988

3. 康立山，孙乐林，陈毓屏. 解数学物理问题的异步并行算法. 北京：科学出版社，1985

（李元香　康立山）

pinfan moshi wajue

频繁模式挖掘（frequent pattern mining）

从数据集中发现频繁模式的过程。**频繁模式**指一个数据集中出现频度超过给定阈值的项集、子序列或者子结构。

频繁模式挖掘最早由 R. Agrawal 等人在针对客户购买行为分析的关联规则挖掘中提出。他们通过对客户超市购买商品数据的挖掘，发现“尿片-啤酒”之类的频繁模式。此后，频繁模式挖掘作为关联规则挖掘的关键步骤，一直是**数据挖掘**领域的重要研究问题，并被扩展到序列模式挖掘（参见**序列挖掘**）（在序列数据集中挖掘子序列模式）、结构模式挖掘（参见**图挖掘**）（在图数据库中挖掘频繁子结构）、相关性挖掘、关联分类以及基于频繁模式的聚类等。

频繁模式挖掘的最早算法是 R. Agrawal 等人提

出的并以他们的姓氏首字母命名的 AIS 算法，而最著名的则是 R. Agrawal 等人稍后提出的 Apriori 算法。该算法的核心依据是：长度为 k 的频繁项集包含的所有长度为 $k-1$ 的子项集必定是频繁的。以此为出发点，通过两个不断循环推进的阶段挖掘数据库中所有频繁项集：①构造候选项集：先扫描数据库，获得长度 $k=1$ 的频繁项集；然后，基于长度 k 的频繁项集，通过连接操作，构造长度为 $k+1$ 的可能频繁项集即候选项集；②验证候选项集：再次扫描数据库，验证候选项集的发生频度是否满足给定支持度。如果找不到长度 $k+1$ 的频繁项集，则挖掘结束；否则进入下一循环。

大量后续研究工作对 Apriori 算法进行了扩展和完善，主要包括如下方面：①更高效的挖掘算法：基于数据集水平划分的挖掘算法、基于取样的挖掘算法、不同长度项集并发挖掘的算法（DIC 算法）以及利用哈希函数统计项集频度的算法等；②无须产生候选项集的挖掘算法：基于 FP 树的挖掘算法及其变种；③最长与闭合频繁模式挖掘；④按列组织的数据集的频繁模式挖掘；⑤分布频繁模式挖掘；⑥并行频繁模式挖掘；⑦增量频繁模式挖掘；⑧带约束的频繁模式挖掘；⑨压缩与近似频繁模式挖掘；⑩高维、多层次频繁模式挖掘等；⑪隐私保护的频繁模式挖掘；⑫非交易数据库频繁模式挖掘，如针对时空数据、文本数据、多媒体数据、流数据、图与网络数据、生物数据等的频繁模式挖掘。

频繁模式挖掘具有广泛的应用，从资源优化到流程规划，从股市行情预测到文本分类与聚类，从关键词提取到软件缺陷发现，从生物模体挖掘到基因调控关系发现等。

频繁模式挖掘本质上是对海量数据的浅层统计分析，挖掘的模式只是对当前数据状态的反映，未必是某种内在的规律。如何从海量数据中高效获取反映数据产生机制的有趣模式依然是频繁模式挖掘面临的主要挑战。

参考文献

1. Agrawal R, Srikant R. Fast algorithm for mining association rules in large databases. In: Proceedings of VLDB 1994. 1994, 487-499

2. Han J, Kamber M, Pei J. Data mining: Concepts and techniques. 3rd ed. Morgan Kaufmann, 2011

（周水庚）

pinfen fuyong

频分复用（frequency division multiplexing, FDM） 按照频率的不同来复用多路信号的技术。在频分复用技术中，信道的物理带宽被分成若干个互不重叠的频段，每路信号占用其中一个频段，在接收端，采用适当的带通滤波器，即可将多路信号分开，从而恢复出传输的原始信号，参见图 1。

频分复用技术适用于模拟信号的复用。

频分复用技术的工作过程如下：

（1）发送端的复用器对各信号进行低通滤波，限制各信号的带宽，避免它们的频谱出现互相混叠；

（2）然后各路信号进行载波调制（也可简单地理解为升频），每路信号被调制到对应的频段，合成后送入复用信道；

（3）在接收端，分别采用不同的中心频率进行带通滤波，分离出各路调制信号，解调后，还原出原始信号。

参考文献

Tanenbaum A S, Wetherall D J. 计算机网络. 北京：清华大学出版社，2012

（袁华）

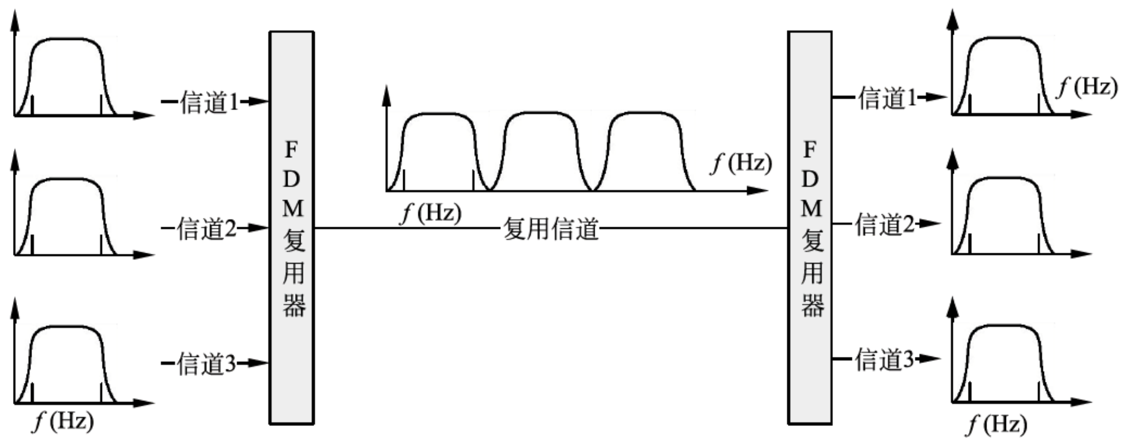


图 1　频分复用技术原理示意图

pinlü tiaozhi jishu

频率调制技术(frequency modulation technology)　频率调制技术根据信息源信号的类型分为数字调频(frequency-shift keying, FSK)和模拟调频(frequency modulation, FM)。

模拟调频是使高频载波信号的频率随调制信号的幅度变化而变化,但载波信号的振幅不变的调制技术。

数字调频,又称频移键控法,指将不同的载波频率f_1、f_2分别对应于调制信号的二进制数值"1"和"0",如下式所示

$$s(t)=\begin{cases}A\cos(2\pi f_1+\phi_1) & (\text{代表编码“0”})\\ A\cos(2\pi f_2+\phi_2) & (\text{代表编码“1”})\end{cases}$$

接收端利用检波器对接收到波形的频率进行区分,并恢复信号中的数字"1"和"0",来完成频率解调。图1为数字调频。

参考文献

1. [日]关清三. 数字调制解调基础. 北京:科学出版社,2002

2. 蒋青,于秀兰. 通信原理. 北京:人民邮电出版社,2008

(董守斌　张凌)

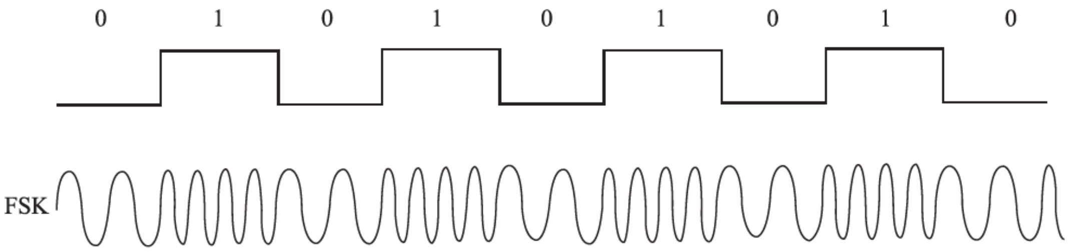


图1　数字调频

pingban jisuanji

平板计算机(tablet personal computer)　一种平板状的小型笔记本计算机(参见**笔记本计算机**)。它以触摸屏作为基本的输入设备,通过手指或触控笔进行操作,摒弃了主要使用键盘或鼠标的传统输入方式。2002年美国微软公司针对平板计算机推出了Windows XP Tablet PC版,平板计算机开始受关注。但是,没有达到真正的流行。比尔·盖茨提出了平板计算机应该是x86架构的无须翻盖、没有键盘、小到足以放入女士手提包,功能完整的个人计算机(PC)。2010年1月苹果公司史蒂夫·乔布斯发布了"iPad"平板计算机。iPad基于ARM架构,不是传统的PC架构。由于iPad创新的设计理念和全新的操作体验,iPad获得空前成功。iPad的畅销激励了平板计算机厂商,他们吸取了iPad的成功经验,掀起了平板计算机热,主要PC厂商都推出了各自品牌的平板计算机产品。

平板计算机大致可以分为三大类:即由微软公司定义的平板个人计算机(tablet PC)和在苹果公司iPad影响下,形成的后PC时代的平板计算机(tablet computer, pad);以及泛指的平板计算机(tablet computer),例如:工业用平板计算机。

平板个人计算机和后PC时代平板计算机(统称平板计算机)的相同之处

(1)平板计算机与笔记本计算机无本质的区别,采用类似的处理器、体系结构、存储设备和输入输出设备。外形和重量介于笔记本计算机和个人数字助理(参见**个人数字助理**)之间。平板计算机的最大特点是采用创新的数字墨水技术和手写输入以及强大的笔输入识别、手势识别、语音识别功能来实现输入。它还支持**键盘**和**鼠标**的传统输入操作,并具有更好的移动性,可以称作是浓缩版的笔记本计算机。平板计算机扩展了PC的使用方式,使用专用的"笔",在计算机上操作,使其像纸和笔的使用一样简单,通过数字墨水技术,可将墨迹(包括文字、标记和图形)存入计算机,呈现了新颖的操作体验。在网络应用方面,平板计算机具有笔记本计算机和**上网本**配置的无线局域网或无线广域网的通信接口。

(2)平板计算机的显示屏即触控屏　大部分平板计算机的触控数位屏采用数位板技术。它允许用户通过手指、触控笔或数字笔来进行操作。屏幕分电阻型和电容型两类,前者具有较高的定位精度,后者具有更快的响应速度。

(3)平板计算机的构成　可分为纯平板型、内置键盘型和混合型三类。

① 纯平板型　将计算机主机与数位液晶屏集成在一起,手写是其主要输入方式,更强调在移动中使用,也可随时通过通用串行总线(USB)端口、红外

接口或其他端口外接键盘或鼠标。

② 内置键盘型　可分为旋转型和滑盖型。旋转型将键盘与计算机主机集成在一起，计算机主机则通过一个巧妙的结构与数位液晶屏紧密连接，数位屏与主机折叠在一起时可当作一台“纯平板计算机”使用。当将数位屏掀起时，该机又可作为一台具有数字墨水和手写输入功能的笔记本计算机。它的屏幕不仅可以进行上下翻折，还可以进行180°的旋转，以便将显示画面展示给计算机旁的其他人员，外观上接近于笔记本计算机。滑盖型平板计算机的结构类似滑盖手机，滑开数位屏幕即可使用内置键盘，其屏幕还可以掀起，采用这种结构的平板计算机数量较少。

③ 混合型　跟“内置键盘型”类似，但混合型平板计算机的键盘是可以分开的，因此既可以把它当作纯平板型，也可以当作内置键盘型使用。

（4）平板计算机的主要优点在于全新的操作体验，对需要站立操作计算机的人员，如新闻播音员；或者休闲时，想躺在沙发或床上浏览计算机的人员，操作者可以一手握着计算机，一手用笔尖或手势和触控屏上的图标，通过触控屏方便地操作计算机。平板计算机的数字墨水技术，使操作者可以把液晶数位板当作纸，用电子笔在屏幕上书写或绘制有创意的艺术作品。由于平板计算机的软件系统中增加了“ink”的数据类型，可将操作者在屏幕上留下的电子墨迹存入系统，并通过输出设备输出，做到了所写即所得，突破了键盘输入的汉字编码与输出的汉字不一致的弊端。艺术家可以方便地将在数位屏上创作的绘画存入计算机。总之，平板计算机除了优异的移动性和便携性之外，新颖的操作体验更是平板计算机的突出优点。

平板个人计算机和后 PC 时代平板计算机的区别

（1）**中央处理器**（以下简称处理器）　根据微软公司的定义，采用 x86 架构处理器是平板个人计算机的基本条件。平板计算机除了采用 x86 架构的处理器以外，包括 iPad 在内的、更多的主流平板计算机采用的是异步响应模式（ARM）架构的处理器。

（2）操作系统　根据微软公司的定义，平板个人计算机必须能够安装 x86 版本的 Windows 系统、Linux 系统或 Mac OS 系统。平板计算机除了配置上述平板个人计算机的三类操作系统外，主流的平板计算机 iPad 安装的是苹果公司为 i 系列产品 iPhone、iPod、iPad 开发的 iOS 操作系统。另一类操作系统是 Google 公司基于 Linux 核心为智能手机、**上网本**开发的 Android 软件平台和操作系统。2011 年 Google 推出了适用于平板计算机的 Android 3.0 蜂巢（Honey Comb）操作系统。平板计算机需要的是轻量级的操作系统，iOS 和 Android 操作系统符合轻量级的操作系统的要求，具有更短的冷启动时间、占用更少的资源，更适合于平板计算机。微软公司 2012 年 10 月推出了适用于超级本（ultrabook）和平板计算机的 Windows 8，Windows 8 除了支持 x86 架构的处理器外，首次破例地支持 ARM 架构的处理器。它是继 Windows 7 之后的新一代操作系统，该系统具有全新的设计理念和更好的续航能力，且启动速度更快、占用内存更少，并兼容 Windows 7 所支持的软件和硬件，促进了平板 PC 与后 PC 时代平板计算机的融合。

（3）开放策略　平板个人计算机沿用了 PC 市场的开放策略，采用开放式的体系结构、标准的硬件总线和软件接口，可安装大量现成的 PC 软件等。而平板计算机的主流产品 iPad 采用的是封闭式的体系结构，不能安装现成的 PC 软件，难以完全替代现有的 PC，这也许是平板计算机需设法解决的问题。

（4）市场定位和操作体验　平板个人计算机定位于商务应用市场，对中央处理器、存储设备等计算机性能指标十分关注。通过电子笔的手写输入，电子墨水和墨迹存储技术改善了人们对计算机的操作体验。平板个人计算机的目标是取代传统的 PC。平板计算机则定位于媒体和移动娱乐应用，对计算机的指标并不关心。通过多点触控（参见**多点触控**）和手势识别技术，操作者使用手势即以一定规律在屏幕上用多个手指的滑动形成的触控轨迹，通过计算机的识别，可大幅度提高计算机的操控能力，简化了计算机的操作过程，为操作者创造了更人性化的操作体验。

工业用平板计算机

工业界俗称一体机。整机性能完善，具有商用平板计算机的性能。产品的技术特点是采用工业主板，它与商用主板的区别在于多品种、非量产，产品型号比较稳定。中央处理器既有采用 x86 架构的，也有选用 RISC 架构的。屏幕、操作系统等与商用平板计算机类似。但是，工业用平板计算机根据应用环境的不同，对可靠性和抗恶劣环境（如防水、防雾、抗振动、高低温冲击等）有不同等级的要求。工业应用需求比较简单单一，性能要求不高，但是要求性能稳定，一般散热量小、无风扇散热。工业平板计

算机的另一个特点就是多数都配合组态软件一起使用,实现工业控制。

总之,工业用平板计算机是一类小批量、多品种、高可靠、高可用的价格相对昂贵的平板计算机产品。

平板计算机集移动商务、移动通信和移动娱乐于一体,注入了手写输入、电子墨迹和手势识别等新技术,开创了更加人性化的操作体验。平板计算机解决了数字化笔记、无纸办公等问题,派生出了**电纸书**等产品。平板计算机的手写输入解决了不能打字但能握笔的残疾用户的困扰,为数码艺术家们提供了一个会变换图像的“本子”,为媒体主持人提供了一个可站立使用,操作简便的“本子”。总之,平板计算机的超一流的操作体验扩大了计算机的使用范围,向更直观、更人性化方面发展。　　(韩承德)

pingfang bijin

平方逼近(approximation in quadratic norm)

用 $\|f-g\|_2=\left[\int_a^b[f(x)-g(x)]^2\omega(x)\mathrm{d}x\right]^{\frac{1}{2}}$ 来度量函数 f 与 g 的距离的逼近(参见**数值逼近**)。

就平方逼近来说,被逼近函数 f 可以属于比连续函数类 $C[a,b]$ 更广的函数类,即所有使 $\int_a^b|f(x)|^2\times\omega(x)\mathrm{d}x$ 存在的 f 之集合,记作 $L_\omega^2[a,b]$。定义 $L_\omega^2[a,b]$ 中两个函数 f 与 g 的内积为

$$(f,g)=\int_a^b f(x)g(x)\omega(x)\mathrm{d}x$$

当 $(f,g)=0$ 时,称 f 与 g 正交。设 $\varphi_0,\varphi_1,\cdots,\varphi_n$ 为 $L_\omega^2[a,b]$ 中一组线性无关的函数,它们的所有线性组合所构成的函数集合记作 φ,则对于每个 $f\in L_\omega^2[a,b]$,在 φ 中的最佳平方逼近

$$S_n(x)=\sum_{k=0}^n C_k\varphi_k(x)$$

存在且唯一,系数 C_k 通过求解 $n+1$ 阶线性代数方程组

$$\sum_{i=0}^n(\varphi_i,\varphi_k)C_i=(f,\varphi_k),\quad k=0,1,\cdots,n$$

而得

$$C_k=\frac{G(\varphi_0,\cdots,\varphi_{k-1},f,\varphi_{k+1},\cdots,\varphi_n)}{G(\varphi_0,\varphi_1,\cdots,\varphi_n)},\quad k=0,1,\cdots,n$$

其中

$$G(\varphi_0,\varphi_1,\cdots,\varphi_n)=\begin{vmatrix}(\varphi_0,\varphi_0) & (\varphi_0,\varphi_1) & \cdots & (\varphi_0,\varphi_n)\\(\varphi_1,\varphi_0) & (\varphi_1,\varphi_1) & \cdots & (\varphi_1,\varphi_n)\\\vdots & \vdots & \vdots & \vdots\\(\varphi_n,\varphi_0) & (\varphi_n,\varphi_1) & \cdots & (\varphi_n,\varphi_n)\end{vmatrix}$$

称为格拉姆行列式。特别地,当 φ_k 为正交函数系时,C_k 可简化为

$$C_k=\frac{(f,\varphi_k)}{(\varphi_k,\varphi_k)},\quad k=0,1,\cdots,n$$

并称为 f 的广义傅里叶系数,而 $S_n(x)=\sum_{k=0}^n C_k\varphi_k(x)$ 称为 f 按正交函数系 $\{\varphi_0,\varphi_1,\cdots,\varphi_n\}$ 的级数展开式,且有最小偏差

$$\|S_n(x)-f(x)\|_2=(\|f\|_2-(S_n,f))^{\frac{1}{2}}$$

当 $\varphi_k(x)$ 为代数多项式时就得到最佳平方逼近多项式。若 $\varphi_k(x)$ 为 $[-1,1]$ 上的勒让德多项式 $\mathrm{P}_k(x)$,则 f 在 $[-1,1]$ 上的最佳平方逼近多项式为

$$S_n(x)=\sum_{k=0}^n\frac{2k+1}{2}(f,\mathrm{P}_k)\mathrm{P}_k(x)$$

这里 $\mathrm{P}_0\equiv1,\mathrm{P}_k(x)=\frac{1}{2^k k!}\frac{\mathrm{d}^k}{\mathrm{d}x^k}[(x^2-1)^k](k=1,2,\cdots,n)$ 为勒让德多项式。

若 $\varphi_k(x)$ 为 $[-1,1]$ 上的切比雪夫多项式 $\mathrm{T}_k(x)$,则 f 在 $[-1,1]$ 上的最佳平方逼近多项式为

$$S_n(x)=\sum_{k=0}^n C_k\mathrm{T}_k(x)$$

其中

$$C_0=\frac{1}{\pi}\int_{-1}^1\frac{f(x)}{\sqrt{1-x^2}}\mathrm{d}x$$

$$C_k=\frac{2}{\pi}\int_{-1}^1\frac{T_k(x)f(x)}{\sqrt{1-x^2}}\mathrm{d}x,\quad k=1,2,\cdots,n$$

$$\mathrm{T}_k(x)=\cos(k\arccos x)$$

它可作为 f 在 $[-1,1]$ 上的近似最佳一致逼近多项式。

平方逼近的计算过程非常方便,故很有实用价值。

参考文献

Davis P J. Interpolation approximation. Waltham, MA: Blaisdell, 1963

(沈毅)

pingjun guzhang jiange shijian

平均故障间隔时间(mean time between failures, MTBF)

指在计算机系统运行的一个较长时间段内,两次相邻故障间隔的平均时间。平均故障间隔时间是衡量**计算机系统可靠性**的一个重要指标,它的值越大表示计算机系统的可靠性越高,可连续正常运行的时间越长。平均故障间隔时间(MTBF)作为一种决策依据已出现近一个世纪,适用于

多种行业,也是计算机界的常用术语。

MTBF 的计算分为"预测"和"估计"两种方法。预测法根据系统设计计算值,通常在产品(系统)生命周期的早期使用,适用于现场数据很少或没有的情形。例如,设计新的产品。如果有大量的现场数据,则应使用估计法。目前,估计法是计算 MTBF 时使用最广泛的方法,其主要原因是这种方法基于现场实际使用的真实产品。这两种方法本质上都是统计型的,只是实际 MTBF 的近似值。没有一种方法是适合于整个行业标准化方法的。

在计算机系统设计中,MTBF 的计算公式为 $MTBF = t/[N(t)+1]$,其中 t 为系统运行的一段较长时间;$N(t)$ 为系统在 $[0,t]$ 时间段内的故障次数。在进行可靠性预测时,可用组成系统的各种成分的失效率来计算,$MTBF = 1/\lambda$,其中 λ 为系统的失效率,是各分系统、设备或器件的固有失效率 λ_i 的总和 $\sum \lambda_i$。每种器件在材料和工艺水平相同的情况下都有相对固定的失效率。该失效率可以在器件的实际使用中得到,也可以通过抽样做加速寿命实验获得。可以按照公式 $\lambda_i = n/NT$ 计算器件平均失效率,其中 n 为失效元件个数,N 为参加运行元件个数,T 为运行时间。如果是通过加速寿命实验获得,还要除以加速因子以求出正常工作环境下的失效率,$\lambda_i = n/NT\sigma$,σ 是加速因子,可根据实验条件或查已有的加速因子表获得。

参考文献

1. Pecht M G, Nash F R. Predicting the reliability of electronic equipment. Proceedings of the IEEE, 1994, 82(7): 992-1004

2. Wendy Torell, Victor Avelar. 平均故障间隔时间:说明和标准,美国可用性研究中心第 78 号白皮书. American Power Conversion, 2004

3. 猪濑·博. 计算机系统的高可靠性技术. 尤国峻,肖俊远,译. 北京:国防工业出版社, 1985

(胡侃　应秋丽　苏振泽)

pingjun xiufu shijian

平均修复时间(mean time to repair, MTTR) 描述产品由故障状态转为工作状态时修理时间的平均值,也叫平均恢复时间。它是产品维修性的一种基本参数。其度量方法为:在规定的条件下和规定的时间内,产品在任一规定的维修级别上,修复性维修总时间与在该级别上被修复产品的故障总数之比。即用如下运算式来表示:

$$MTTR = \frac{\sum_{i=1}^{n} t_{[ri]}}{\sum_{i=1}^{n} r_{[i]}}$$

式中,$t_{[ri]}$ 表示使用期内第 i 台受试产品出现故障后修复时间,$r_{[i]}$ 为使用期内第 i 台受试产品出现故障的次数。

平均修复时间源自于 IEC 61508 中的平均维护时间(mean time to repair),目的是为了清楚界定术语中的时间概念,MTTR 是随机变量修复时间的期望值。

通常,MTBF(mean time between failures,平均故障间隔时间)和 MTTR(mean time to repair,平均修复时间)和 MTTF(mean time to failure,平均无故障时间)是评价系统可靠性和可用性的三个主要技术指标(参见**计算机系统可靠性**)。

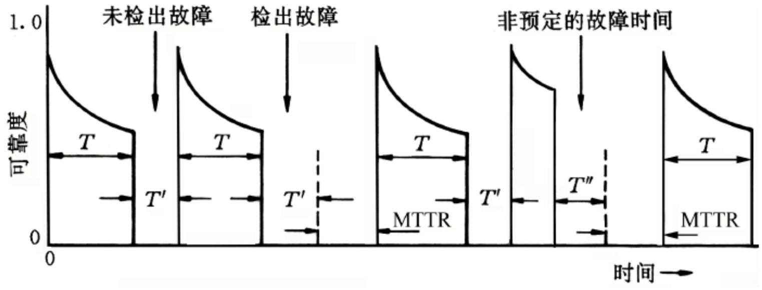
平均故障间隔时间是新产品在规定的工作环境条件下从开始工作到出现第一个故障的时间的平均值。平均故障间隔时间越长表示可靠性越高,正确工作能力越强。

平均修复时间,是从故障发生到恢复所需的平均时间,广义的 MTTR 还涉及备件管理、客户服务等,是设备维护的一项重要指标。平均修复时间越短表示易恢复性越好。

平均无故障时间是系统平均能够正常运行多长时间才发生一次故障。系统的可靠性越高,平均无故障时间越长。

计算机在使用过程中要经受检验、运行、故障、修理等周期,一个经维护的系统的周期性能,如图 1 所示。修复时间是从发现 1 次失效到修复至正常状态所需时间,它包含诊断、故障定位、修理实施等时间。平均修复时间是修复时间的平均值。

高可靠性和高可用性是相辅相成的,但不可以互换。可靠性是指系统或组件在规定的条件下按照指定的时间实现其应实现功能的能力。可用性是指系统或组件在需要使用时正常使用的可能性。MTTR 影响可用性,但不影响可靠性。MTTR 越长,系统情况越差。简言之,如果系统从故障中修复所需的时间越长,系统的可用性就越低。可用性 $= \frac{MTBF}{(MTBF + MTTR)}$ 说明了 MTBF 和 MTTR 如何影响系统的整体可用性。随着 MTBF 的增大,可用性会提高;随着 MTTR 的增大,可用性将下降。



T —— 容许的最大“可用”时间　T' —— 由于预防性维护的停机时间

T'' —— 非预定的诊断　MTTR —— 平均修复时间

图 1　一个经维护的系统的周期性能

参考文献

1. IEEE Standard Computer Dictionary, 610.12, 1990

2. European Power Supply Manufacturers Association. Guidelines to Understanding Reliability Prediction, 2005

3. Schneeweiss W G. Computing Failure Frequency, MTBF & MTTR via Mixed Products of Availabilities and Unavailabilities. IEEE Trans. on Reliability, 1981

（郑然）

pingmian tu

平面图（planar graph） 能够在平面上画出，且边不在非顶点处相交的**无向图**。在平面上画出的边不在非顶点处相交的平面图 G 的图形称为 G 的平面表示。在平面图 G 的一个平面表示中，以 G 的边为边界的连通区域称为 G 的该平面表示的面。一个平面图可以有多个不同的平面表示，但其任何平面表示的面的数目均一样。这可以用著名的欧拉公式进行计算；若 n 阶平面图 G 有 m 条边和 k 个分支，则 G 的一个平面表示的面数 $f = m - n + k + 1$。在无向图 G 的任一条边 e 上插入一个度为 2 的顶点，将 e 一分为二，或者从 G 中删去一个度为 2 的顶点 v，将与 v 关联的两条边合二为一，可得到图 G'，此时称 G 与 G' 同胚。波兰数学家 Kuratowski 给出了判断一个图是否为平面图的准则，即著名的库拉托夫斯基定理：无向图 G 是平面图当且仅当 G 没有同胚于图 1 或图 2 的子图（参见**图论**）。

给定平面图 G 的一个平面表示，在其每个面内找一顶点，若两个面有 m 条公共边，则用 m 条线连接这两个面内取定的顶点，并使其分别与 m 条公共边相交。由这些顶点和连线组成了边不在非顶点处

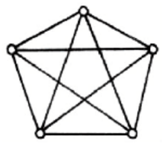


图　1

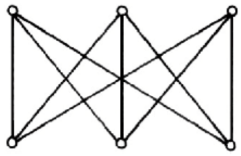


图　2

相交的图形。称以该图形为其平面表示的平面图为 G 的对偶图。

（张强）

pusu beiyesi fenleiqi

朴素贝叶斯分类器（Naïve Bayes classifier）

在朴素的条件独立性假设的基础上运用贝叶斯定理而得到的分类器。条件独立性假设是指，在类变量的条件下，一个特征的概率分布与其他任何特征无关。具体地，对于特征向量 $\boldsymbol{X} = [X_1, \cdots, X_d]^{\mathrm{T}}$，其在类变量 C 的条件下出现的概率为

$$P(\boldsymbol{X} \mid C) = \prod_{i=1}^{d} P(X_i \mid C). \qquad (1)$$

朴素贝叶斯分类器的条件独立性假设成功地将多元随机变量的参数估计转化成了多个相互独立的一维随机变量的参数估计，大大降低了模型的复杂度。当 X_i 取离散值时，可以假定每一维特征变量服从伯努利分布或二项式分布。一般地，当 X_i 取连续值时，通常假定其服从高斯分布或非参数形式的概率分布；另外，也可以通过离散化步骤将 X_i 转化为离散值估计概率分布（用直方图表示）。

假设分类问题中类别的数目为 K，那么通过在给定类标号的样本集上用极大似然估计法（maximum likelihood estimation）可以计算出模型参数的估计值 $\hat{P}(X_i | C_k)$ 和 $\hat{P}(C_k)$，其中 $i = 1, 2, \cdots, d, k = 1, 2, \cdots, K$。根据贝叶斯决策理论，朴素贝叶斯分类器将 X 分给后验概率 $\hat{P}(C_k | X)$ 最大的那一类。在条件

独立性假设下,根据贝叶斯公式计算后验概率为

$$\hat{P}(C_k \mid X) = \frac{\hat{P}(C_k)\prod_{i=1}^{d}\hat{P}(X_i \mid C_k)}{\hat{P}(X)} = \frac{\hat{P}(C_k)\prod_{i=1}^{d}\hat{P}(X_i \mid C_k)}{\sum_{k=1}^{K}\hat{P}(C_k)\prod_{i=1}^{d}\hat{P}(X_i \mid C_k)},$$

$$k = 1,2,\cdots K. \tag{2}$$

显然,在满足条件独立性假设时,朴素贝叶斯分类器是最优的贝叶斯分类器,即决策风险最小或0-1 损失情况下分类错误率最小。实际上,现实生活中的分类问题很难严格满足条件独立性假设,在该假设近似满足时,朴素贝叶斯分类器也能提供优越的分类性能。有研究证明,即使在条件独立性假设不满足时,0-1 损失情况下朴素贝叶斯分类器仍然可能是最优的,因为即使根据公式(2)近似计算的后验概率不准确,只要近似后验概率最大的类别和真实后验概率最大的类别一致,分类就是最优的。

然而,在对多种不同分类器的实验比较中,研究者们发现许多分类器如支持向量机(support vector machine)、AdaBoost 分类器的分类性能会超出朴素贝叶斯分类器。这表明在实际的复杂问题下,属性变量之间的依存关系不能完全被忽略,因此需要更为强大的工具——**贝叶斯网**(Bayesian network)来刻画这种依存关系。一些限制性贝叶斯网在适当放宽朴素贝叶斯的条件独立性假设之后明显提升了分类性能,如 TAN (tree augmented naive Bayes)和 AODE (averaged one-dependence estimators)。TAN 假定任意 X_i 除依赖类变量 C 以外最多依赖一个属性变量,而 AODE 计算 d 次概率估计的平均值作为最终估计,在第 i 次估计中所有变量均依赖类变量 C 和属性变量 X_i。

参考文献

1. Bishop C M. Pattern recognition and machine learning. New York: Springer, 2006

2. Domingos P, Pazzani M. On the optimality of the simple Bayesian classifier under zero-one loss. Machine Learning, 1997, 29: 103-137

3. Lewis D D. Naive (Bayes) at forty: the independence assumption in information retrieval. In: ECML'98, Tenth European Conference on Machine Learning, LNCS 1398, 1998, 4-15

(靳小波 刘成林)

pushi jisuan

普适计算(pervasive/ubiquitous computing)

一种继主机计算和桌面计算之后的新型计算模式。它通过嵌入在生活环境与日常工具中的计算机,把信息空间与人们生活的物理空间融合成为一个整体,通过和谐的人机交互,使用户能随时随地访问信息资源。这种模式追求以人为本,要求计算机能感知周围环境的变化,并主动根据用户的隐式或显式需求做出响应。

普适计算的概念最早由 Xerox PARC 首席科学家 Mark Weiser 于 1991 年提出,从 20 世纪 90 年代后期开始这一概念得到广泛关注和接受,无论学术界、政府部门和产业界,均在推动这一技术的发展。

普适计算具有以下特点:

1. 透明性

即通过将相互连通的计算设备隐藏于物理环境,使计算机本身从人们的视线里消失,从而实现信息世界与物理世界的无缝融合。这种融合使得用户在访问信息世界的时候,无须花费更多的注意力,甚至是在用户根本注意不到的情况下,发生了对信息世界的访问。这种对信息世界的访问是一种蕴涵式的交互。为了实现蕴涵式的交互,需要和谐**人机交互技术**的支撑。和谐的人机交互其最终目标是人与计算机的交互如人与人的交互一样智能,如人与物理世界的交互一样自然。

2. 计算无所不在

用户可以随时随地借助各种隐藏于环境中的计算机或者携带的计算设备进入信息世界,获取信息资源。无所不在的计算要求计算设备被部署于整个物理空间,并且计算设备通过通信网络互相连通。因此,无所不在意味着设备无所不在,通信无所不在,从而实现普适计算服务的无所不在。

3. 以人为本的服务

传统的计算要求用户去迎合计算机,如用户必须熟悉计算机的使用方式,才能正确获得计算服务;而普适计算让计算机迎合用户,这种迎合不仅体现在人机交互的形式,更体现在计算所提供的服务。以人为本的计算依赖于计算对用户周围环境的感知,以及对用户状态及需求的掌握。环境信息以及用户状态、需求等各类信息组合在一起,形成计算所需的情境上下文,因此,情境计算是实现普适计算的一项支撑技术,它在本质上决定了普适计算如何向用户提供以人为中心的智能化服务。

普适计算是计算、通信和数字技术等多种技术

的融合,其涉及的研究领域包括分布式计算、**移动计算**、人机交互、**人工智能**、嵌入式系统、感知网络以及信息融合等。

参考文献

徐光祐,史元春,谢伟凯．普适计算．计算机学报,2003,26(9)　　(过敏意)

pubu moxing

瀑布模型(waterfall model)　一种软件开发模型,该模型将软件生存周期的各项活动规定为依固定顺序联接的若干阶段工作,形如瀑布流水,最终得到软件产品。

瀑布模型可追溯到50年代末期,当时人们已感到必须先确认"做什么",才能编制程序将其实现,即使是比较简单的小型问题也不例外。最简单的两级瀑布模型如图1所示。

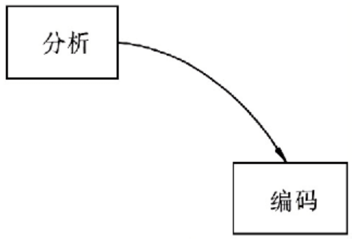


图1　两级瀑布模型

对于较大项目,问题更加复杂,两级模型已不能满足软件开发的实际需要。一个更精确的软件开发步骤可按需要解决问题的顺序依次为:做什么—如何做—制作—检测—使用。于是一个反映软件过程的基本框架如图2所示。该图表明,首先应给出软件的目标,确定要做什么;然后要决定如何达到这一目标,给出策略、方法和步骤;继而加以实施,制作出所要的软件;经过适当的检测,判定符合初始目标以后,方可投入运行和使用。可以说这是瀑布模型的雏型。

1970年W. Royce首先将这一模型精确化,提出

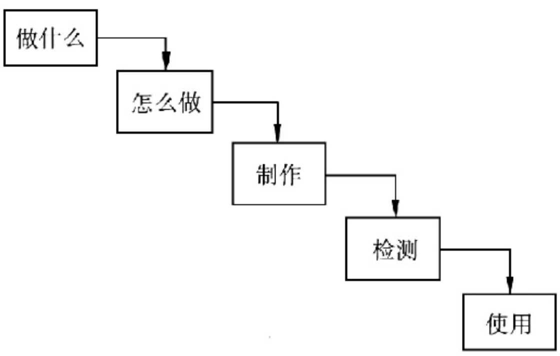


图2　瀑布模型雏型

了具有多个开发阶段的瀑布模型,如图3所示(不包括几个向上的虚线箭头)。这一模型规定了开发各阶段的活动为:提出系统需求、提出软件需求、需求分析、设计、编码、测试和运作。并且还规定了自上而下相互衔接的固定顺序。因而构成了熟知的瀑布模型。然而实践表明,各开发阶段间的关系并非完全是自上而下的线性图式。实际情况是,每个开发阶段均应具有以下特征:

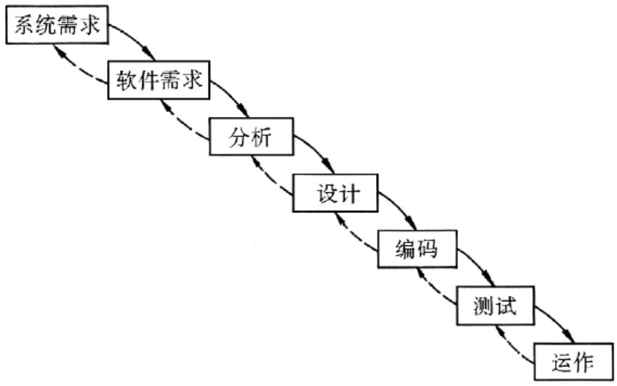


图3　瀑布模型

(1)从上一阶段接受本阶段工作的对象,作为输入;

(2)对上述输入实施本阶段的活动;

(3)给出本阶段的工作成果,作为输出传入下一阶段;

(4)对本阶段工作进行评审,若本阶段工作得到确认,则继续下阶段工作,否则返回前一阶段,甚至更前阶段。

为表达向前阶段的反馈,在图3中增加了虚线表示的箭头,从而构成了具有反馈回路的瀑布模型。

瀑布模型有着不同形式的变种。比如,另一常见的具有反馈回路的瀑布模型包含的7个阶段是:可行性研究、需求分析和规约、设计和规约、编码和单元测试、集成测试和系统测试、交付、维护。不同形式瀑布模型的变种彼此并无本质差别,选择何种形式由软件项目特性及开发机构决定。

许多采用瀑布模型的开发机构为有效地组织实施,制定了软件开发规范或开发标准。其中明确规定了各个开发阶段应交付的产品,这就为严格控制软件开发项目的进程,最终按时交付产品以及保证软件产品的质量创造了有利条件。

参考文献

Carlo G, et al. Fundamentals of software engineering. Prentice Hall,1991　　(郑人杰)

Q

qiye hucaozuo

企业互操作(enterprise interoperability, EI) 指企业内、外部的组织与其他组织之间协同工作和交换共享信息及资源(文件、数据、知识、服务、应用业务及过程、软件等)的一种能力。企业互操作技术是指利用信息技术解决企业组织内、外部的异构应用、软件、服务、系统的交互、协同与集成问题的共性技术。从计算机软件角度来看,互操作是两个或多个软件系统交换信息并使用所交换信息的能力,即指两个可以互操作的软件可以不经过特别的接口设计而建立跨平台的交换信息和服务渠道,很好地协同工作。扩展到企业范围,互操作则涉及了不同企业组织之间在数据、服务、过程、业务不同层次上的互操作。

企业互操作技术是一个跨学科的综合性技术,涉及企业互操作的概念、表示、架构、知识、实现、工具、平台等方面;根据欧盟政府企业互操作研究路线图和近年来的企业互操作研究进展,可以归纳出其主要技术内容包括企业互操作技术体系结构与平台、企业互操作建模技术、企业互操作知识本体论等。

(1) 企业互操作技术体系结构与平台 为利用信息技术实现企业互操作提供系统体系结构、支撑工具和平台,其主要研究内容包括未来互联网环境下企业系统 FInES 体系结构、企业互操作体系结构(EIF)、模型驱动的互操作技术(MDI)、互操作服务实用工具(ISU)、企业互操作与协同平台、企业互操作技术标准等。

(2) 企业互操作建模技术 用于描述企业组织间在数据、服务、过程、业务不同层次上互操作的交互内容、方式与过程,其主要研究内容包括互操作需求定义、企业模型(数据、服务、过程、业务等模型)与建模方法、跨组织互操作业务过程建模方法、互操作过程模型语义、企业模型转换及一致性检查技术、异构数据模型互操作方法、模型驱动的互操作体系结构建模方法、企业互操作方法论等。

(3) 企业互操作知识本体论 为明确定义和标识企业互操作的内容和语义提供知识表示方法及语义描述,其主要研究内容包括企业互操作语义本体论、企业业务知识获取方法与知识表达、互操作业务语义标注及转换、面向知识的语义互操作、面向企业互操作的知识创建/管理/协作支持工具等。

企业互操作技术在各种类型的企业及其协作中有着广泛的应用,其应用范围不仅涉及各行业的制造型企业,还涉及非制造型企业与政府机构等组织之间的协作,有效地帮助了企业提高其业务灵活性,降低企业协作障碍与互操作成本,提高企业开发新商业机遇的能力,提高企业的国际竞争力。企业互操作技术在信息化与工业化融合中也发挥着重要作用。

企业互操作技术的主要发展趋势为:①网络化 随着企业内外部网络环境的变化和对企业系统互操作、开放性和适应性提出的新要求,人们在致力于探索"未来互联网环境下企业系统 FInES"的新型互操作技术、云计算环境下的企业互操作技术。②模型化 为支持从企业业务到技术的多层次互操作,模型驱动的技术框架将成为实现企业互操作的主要方式。③智能化 为了应对企业互操作问题的日益复杂化,对于面向互操作语义的协作知识表示与处理将备受关注,本体论将对面向知识的协作和语义互操作产生更大影响。④协同化 为进一步支持企业间协作,跨企业虚拟组织(VO)及其互操作信息、知识和服务共享将是企业协同的新形态。⑤服务化 随着信息技术的服务化趋势,服务将成为提供互操作能力的主要形式;软件即服务(SaaS)、互操作即服务(IooS)将成为企业间实时信息共享和协作服务的新型规范体系结构。⑥科学化 随着企业互操作技术的深入,人们开始探究企业互操作的科学基础,通过吸纳、综合和扩展其他成熟或新兴科学领域(系统科学、信息科学、网络科学、服务科学、经济学、社会学等)的理论成果,逐步形成企业互操作的科学理论体系。

参考文献

1. Doumeingts G, Mller J, et al. Enterprise interoperability: new challenges and approaches. Springer, 2007

2. Vernadat F B. Interoperable enterprise systems: principles, concepts, and methods. Annual Reviews in Control, 2007, 31(1): 137-145

3. Chen D, Vernadat F B. Enterprise interoperability: a standardization view. In: Proceedings of the IFIP TC5/WG5.12 International Conference on Enterprise Integration and Modeling Technique: Enterprise Inter-and Intra-Organizational Integration: Building International Consensus. Kluwer, B. V., 2003

（徐晓飞　刘晓烽）

qiye yingyong jicheng zhongjianjian

企业应用集成中间件(enterprise application integrator,EAI)　支持企业数据资源共享、应用系统互通和业务流程协同的**分布计算中间件**。

在市场竞争日趋激烈、信息技术飞速发展的今天,企业需要将分散在各个部门的业务应用系统和数据源集成在一起,构建各种企业级应用软件系统,以实现企业的整体业务目标,并对业务变化和新的机遇进行快速响应,获取更大的竞争优势。EAI 中间件能够有效简化企业应用集成过程、降低集成代价、提升集成时效,并通过各个数据源的整合,提升企业数据的精确性,降低数据冗余,并减少数据分布在不同数据源中导致的不一致性。在目前复杂的企业信息环境中, EAI 中间件以松散耦合方式连接异构的系统和操作环境,能够较好适应需求变化,并且在部分失效时,保证最大限度的系统弹性和健壮性。

沿　革

企业应用集成是伴随着社会信息化的发展而逐渐发展起来的。20 世纪 60～70 年代,计算机主要用于科学计算、集中式事务处理和工业控制,由于金融、海关、制造业及国际贸易等应用系统之间数据交换的需求,出现了企业应用集成思想的萌芽,发展形成了电子数据交换的技术、标准和应用。80～90 年代,随着网络和个人计算机的兴起,企业应用软件系统向部门级拓展,特别是,随着互联网的兴起和电子商务的出现,企业开始采用**消息中间件**、分布对象中间件、业务流程集成中间件或其他特定的技术构建企业信息基础架构,对遗留应用制定接口规范,建立信息资源的企业级视图,以实现信息资源的规范化、集中化和综合利用,并支持业务流程自动化,以适应业务环境变化的需求。同重新构造企业信息系统相比较,中间件提供了一种集成应用的更为经济、便捷的手段。进入 21 世纪,随着企业应用集成研究的深入,逐渐形成了对企业应用集成中间件的整体认识,并从数据层、应用代码层、业务流程层和展现层开始了 EAI 中间件技术体系的全面整合,产生了一系列 EAI 套件产品。目前,比较典型的企业应用集成中间件产品有 IBM WebSphere Business Integration 系列, BEA WebLogic Integration, Tibco, webMethod, Microsoft BizTalk,以及多种国产的 EAI 中间件套件。

主要内容

EAI 中间件由支持数据的抽取、转换、加载的数据集成中间件或工具,支持分布式应用代码之间互联互通互操作的应用代码集成中间件,支持业务流程的建模、编排、运行和管理监控的工作流和业务流程集成中间件以及支持展现逻辑的动态组合、统一访问及个性化服务并适应 Web 应用的门户集成中间件等组成。EAI 中间件的技术体系是其内含的各种中间件的有机整合,通常包含应用层、应用适配层、消息代理层和业务流程层四个层次,其基本结构如图 1 所示。

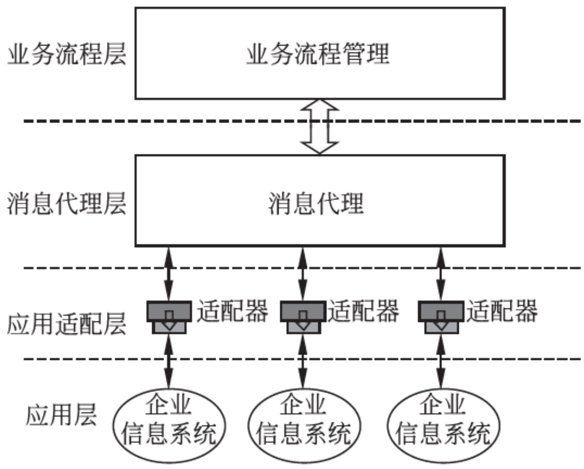


图 1　企业应用集成中间件技术体系的基本结构

(1) 应用层　由各类遗留或新建的企业信息系统(EIS)构成。EIS 是指企业在信息化建设的不同阶段基于不同技术购买或自主开发的应用系统,包括传统客户-服务器模式的管理信息系统、数据库管理系统和企业资源规划系统(ERP)等,它们为企业日常事务管理以及决策支持提供基本的信息支持。

(2) 适配层　由实现数据转换、应用接口的各类应用适配器构成。其功能主要是在异构的 EIS 与消息代理之间架起互联互通互操作的桥梁,即在特殊的 EIS 数据格式、调用访问接口与统一的消息代

理数据格式之间实现相互转换。

（3）代理层　由消息代理构成。通过消息代理的统一管理和传输，各个应用系统之间可以以消息形式向其他感兴趣的应用系统发布信息，或向其他应用系统订阅自己感兴趣的信息，并在传输过程中保证消息的一致性和传输质量。

（4）业务流程层　由一组实现业务流程的定义、组装、装载和运行监控的软件工具构成。业务流程，指按照特定的业务需求而将所有参与相应业务的活动依它们的时序和逻辑关系连接起来的业务功能序列。在EAI中间件中，所有集成活动都通过业务流程定义，并按业务流程驱动。

企业应用集成中间件的工作原理为：在业务流程的驱动下，业务流程引擎与消息代理交互，激活相关应用适配器采集企业信息系统的数据或调用相关接口并进行格式转换，以统一的消息形式提交给消息代理。该消息的订阅者从消息代理中获取所需要的消息，进行相关处理。这样循环往复，实现不同企业信息系统中的数据集成、应用互通和流程整合。

关键技术

（1）适配器　旨在实现企业信息系统中的数据与消息代理中消息格式的相互转换，为企业信息系统之间互连提供可重用、统一的接口。企业应用系统之间的通信方式包括套接字、消息传递、远程过程调用、对象请求代理、HTTP和FTP等，而数据库系统之间常见的连接规范有ODBC、JDBC等，EAI通过对所有这些连接协议的包装，将原有的每个数据库系统、应用系统和网络服务构件都封装成对应于它的一个标准的适配器。每一个适配器只与消息代理相连，彼此之间不直接发生关系。于是，企业信息系统之间也就不直接相连，它们之间的消息传输均由其适配器通过消息代理实现。

（2）消息代理　基于**发布/订阅模型**的一种接发和管理消息的机制。在消息代理中，消息的发送方称为出版者，消息的接收方称为订阅者，不同的消息一般通过不同的主题进行区分。出版者向消息代理出版其他应用系统感兴趣的消息，而订阅者从消息代理接收自己感兴趣的消息，出版者和订阅者之间通过消息代理进行关联。消息代理中的消息通常采用XML统一表示，消息代理涉及的主要技术包括消息队列、消息路由、消息永久存储、事件服务、名字服务、通告服务、安全服务和事务服务等。

（3）业务流程管理　由业务流程建模工具、业务流程引擎和业务流程监控管理工具等组成。①业务流程建模工具　对业务流程进行设计建模并对模型进行集中管理的工具。流程建模通常采用图形化方式来描述业务流程各个环节的业务活动及资源约束，并刻画流程变化的业务规则。②业务流程引擎　通过维护业务流程、活动实例等控制状态信息，实现业务流程解释执行、业务流程实例的生命周期管理、流程实例执行过程控制和业务活动的执行顺序安排。③业务流程监控管理工具　对系统中的核心业务流程或关键活动的执行过程进行监控管理的工具。通过对业务流程的分级和分类，在流程运行过程中监查特定流程实例或关键活动，从而可以实时跟踪业务流程的动态执行过程，并对业务流程执行情况进行统计分析。

发展趋势

随着EAI以及Web服务的发展，出现了面向服务体系架构（SOA）、企业服务总线（ESB）和实时企业（RTE）等概念和技术，它们将引领未来EAI中间件技术的整体走向，EAI中间件将向服务化、智能化、实时化发展。其中，服务化是指EAI中间件将与SOA和Web服务等技术相融合，其核心的消息代理机制将向服务代理和企业服务总线过渡；智能化是指在数据、应用、业务流程已经有效集成的基础上，决策支持系统等智能化应用将成为企业应用集成中间件的一个重点支持方向，同时，灵活的业务规则引擎、消息智能路由和集成策略管理机制将在EAI中间件中得到进一步的丰富和完善；实时化是指EAI中间件将进一步消除企业内部存在的各种信息孤岛，实现企业信息资源的无缝衔接，保障企业各类业务活动所需信息资源的新鲜性和实时性，消除关键业务流程管理与执行中的延迟，并对业务流程进行动态优化调整，以即时响应市场变化，提升企业的整体竞争力。

参考文献

1. Linthicum D S. Enterprise application integration. Addison-Wesley Professional, 1999

2. Chappell D A. Enterprise service bus. O'Reilly Media, 2004

（刘江宁）

qiye ziyuan jihua

企业资源计划（enterprise resource planning，ERP）　一种基于系统化管理思想和信息技术的企业现代管理模式与技术。ERP按照市场需

求和企业经营目标,借助信息技术实现企业物流、资金流和信息流的集成,对企业资源进行合理调配,为企业决策层及员工提供管理与决策手段,实现企业经济效益最大化。ERP 既是一种先进的企业管理思想,又是一种综合的企业集成管理系统。体现 ERP 管理思想的企业集成管理软件也称作 ERP 软件。

企业资源计划(ERP)是在物料需求计划(materials requirement planning, MRP)和制造资源计划(manufacturing resource planning, MRP Ⅱ)的基础上发展而来的。20 世纪 60 年代,美国 IBM 公司最早提出并实现了物料需求计划(MRP)。MRP 是一种基于物料管理和库存计划的生产管理思想;它依据最终产品的需求数量和交货期、提前期和批量政策等信息,根据反映产品结构的物料清单(bill of material, BOM),计算零部件乃至原料的需求数量和需求日期,再依据库存及生产信息对采购对象和自制对象做出计划安排,以降低库存和生产管理成本。70—80 年代,物料需求计划(MRP)逐步发展为制造资源计划(MRP Ⅱ);MRP Ⅱ 在物料需求计划 MRP 的基础上,将对企业内部的各类制造资源(如物料、设备、人力、资金、信息等)进行总体计划和优化管理,形成集成的制造资源管理系统,在企业有限制造资源条件下,取得更大经济效益。90 年代,面对经济全球化和全球信息化环境,美国著名的信息分析咨询公司 Gartner Group Inc. 在总结 MRP Ⅱ 的基础上,借助于供应链管理的思想,提出了 ERP 的概念,将制造资源管理扩展为企业资源管理,即除物料、能力和资金资源的管理外,还包括供应商、分销商、运输商和服务商等社会化资源的管理。与 MRP Ⅱ 相比,ERP 除了功能更为强大和管理的企业资源更多之外,它更加面向全球化市场,涉及了企业内部管理与企业间的供应链管理,所采用的计算机平台系统也更加先进。ERP 软件系统已成为主流的制造企业综合集成经营管理软件系统。

ERP 技术的主要内容包括 ERP 管理模式与技术、ERP 软件系统、ERP 软件开发技术、ERP 实施方法、ERP 标准化技术等方面。

ERP 是一种先进的企业管理模式。它以面向交货期的时间主线管理(如 MRP Ⅱ)为基础,将物料资源、能力资源、资金资源和社会化资源进行集成化管理,进一步融合了精益生产(LP)、准时生产(JIT)、业务过程重组(BPR)、供应链管理(SCM)、客户关系管理(CRM)、敏捷制造(AM)等强化产品制造上、下游供应链协作的先进管理模式,以求最大限度地利用企业内外部资源获取最大化的经济效益。它还强调集成协同管理,既要实现企业内部集成(包括产品设计、制造和管理的集成;制造管理、物流管理和资金管理的集成;计划、调度控制和执行的集成等),又要实现企业与外部的集成,即企业与供应链上下游所有合作伙伴的集成。

ERP 软件系统按功能体系划分,主要包括:制造管理、物流管理、财务管理、资源管理、供应链管理、质量管理等部分。①制造管理:主要围绕制造过程进行制造与能力数据管理、生产计划与调度、车间管理与控制等,可支持如连续流程制造、离散制造等多种生产方式的管理;②物流管理:主要负责对企业的各类物料的采购、库存、销售进行有序管理与控制,为企业的经营生产管理提供保障;③财务管理:将生产、采购、销售等活动的信息计入财务系统,并进行财务执行过程中资金流的优化计划与控制;④资源管理:将与生产制造相关的能力资源进行集成管理,提高资源可用性和利用率;⑤供应链管理:负责企业供应链上下游间的合作伙伴及其相关物流管理,保证企业获取原材料,再将原材料转化为间接产品和最终产品,并将产品分送至顾客手中;⑥质量管理:支持企业全面质量管理和质量过程控制。

为了使 ERP 软件系统能够满足企业资源管理的功能与性能上的需要,并适应企业经营管理业务的不断变化,人们越来越多地采用了先进软件开发技术与系统平台来实现 ERP 系统,主要包括面向企业业务动态变化的企业建模技术及模型驱动体系结构(MDA)、支持 ERP 快速实施和系统动态可变性的软构件平台技术、面向系统可集成性的企业服务总线及其新型软件架构、支持企业业务过程重构的工作流管理技术、面向智能生产计划与控制的知识管理技术、面向企业领导决策的数据挖掘与联机分析(OLAP)技术、面向多种业务灵活管理的智能报表技术等。

ERP 的应用实施方法也是 ERP 技术的重要方面。ERP 系统的实施和应用是一个系统工程,涉及企业业务需求分析、业务过程优化、企业管理组织调整、企业数据整理与装载、ERP 软件开发/修改/安装/调试与运行/维护等方面。ERP 实施还是人们对新的企业管理理念的认识和转变、企业业务的梳理和规范化,因此,需要有效的规划和控制。

ERP 的标准化技术是确保 ERP 成功实施的重

要保障。ERP 的标准化涉及 ERP 功能的标准化、企业管理业务过程标准化、企业管理业务数据标准化、制造数据及其编码标准化、管理与控制计量标准、ERP 实施规范等。

由于 ERP 代表着现代企业先进管理模式和先进技术,并广泛适用于多个行业和各种类型、各种规模的企业。ERP 的应用不仅提高了企业管理和信息化水平,还给企业带来巨大的经济效益,已经成为企业现代化管理升级和企业信息化的必要手段。

ERP 技术主要呈现以下发展趋势:①集成化 随着企业信息系统数量与种类的增多,应用软件的互操作与集成技术受到前所未有的重视;ERP 通过企业应用集成平台(EAI)或企业服务总线(ESB)等新一代网络化企业集成平台实现与**产品生命周期管理系统**(PLM)、制造执行系统(MES)、办公自动化系统(OA)以及其他企业信息系统的深度集成,支持企业的产品设计、生产、企业管理、营销的产品生命周期的全过程和企业经营业务的全方面管理。②协同化 由于全球化企业经营与协作引发企业协同管理新模式,使得 ERP 更加面向全球化市场,支持企业内部与企业间的业务协同,并将形成基于新一代**互联网**和**云计算**环境的跨企业协同资源计划管理系统(CORP)。③行业化 由于越来越多的企业要求行业性与企业化的 ERP 解决方案,要求 ERP 管理业务及其软件进一步行业细分,行业参考模型驱动和可剪裁的 ERP 软件系统将更加受到欢迎。④动态化 由于 ERP 应用企业的业务或组织经常性变化,ERP 系统将具有更强的动态演化和重构扩展能力。⑤智能化 企业知识管理、数据挖掘与联机分析(OLAP)技术等将不断提高 ERP 系统的商务智能分析与决策能力。⑥服务化 随着企业生产经营管理方式以及企业信息系统的服务化,基于服务的 ERP 系统也将出现:面向服务的体系架构(SOA)、软件即服务(SaaS)平台、云计算环境等将对未来 ERP 的系统架构和实现方式产生重要影响。

参考文献

1. 陈启申. 供需链管理与企业资源计划. 北京:企业管理出版社,2001

2. 徐晓飞. 现代企业资源计划与管理. 北京:经济出版社,1999

3. 徐晓飞,田雨华,薛劲松. 计算机集成制造系统 CIMS 知识新解. 北京:兵器工业出版社,2000

4. Cox J F, Blackstone J H. APICS 辞典. 11 版. 中译版. 李秉光,霍艳芳,等译. APICS,2008

(徐晓飞 战德臣)

qifashi sousuo

启发式搜索(heuristic search) 一种利用与待解问题有关的信息(称为启发信息),对搜索路径的走向给予一定约束或选择的搜索方法。

搜索方法的目标是要在与问题有关的状态空间或图表示中,根据已知的初始状态(起始节点)、目标状态(满足目标状态描述的节点)以及从一种状态(节点)转换到另一种状态(节点)所允许的操作或算符,寻找一条从初始状态达到目标状态的途径。绝大多数问题求解技术最终都归结为状态空间或图的搜索问题。

一般来说,不同的问题求解类型需要不同的搜索策略。根据问题求解的任务和问题本身所存在的解的情况,问题求解可分为三种类型。一是问题只有唯一解或有多个解,但它们均处于同等地位,不涉及寻找最优解。这类问题要求搜索方法尽可能地减少搜索次数并保证完全性,即问题存在解的话,搜索一定能成功并找到问题的解。定理证明所面临的就是这类问题。二是问题有多个解,问题求解的目的是寻求其最优解。在问题的规模不太大,复杂性不甚高的情况下,这是可以做到的,但对大多数这类问题来说,需利用某些启发信息以提高搜索效率。A^* 和 AO^* 等启发式搜索算法所要解决的就是这一类问题。第三类与第二类相似,但问题是 NP 难解的(参见 **NP 完全性理论**)。在现实的存储资源和时间条件下很难或根本得不到最优解。同时,对于诸如推销员旅行问题等具体应用,令人满意的解也并非一定要最优解。因而在求解这类问题时可以放弃最优解而研究各种更加实用有效的启发式搜索方法。

20 世纪 50 年代末期,A. Newell,J. C. Shaw 和 H. A. Simon 开始研究启发式搜索。60 年代中期以后,随着计算机,尤其是**人工智能**应用领域的不断扩大,NP 难解性问题又长期得不到解决,因而启发式搜索的研究越来越引起人们的重视与兴趣,并且取得了一批引人瞩目的成果。如 J. Doran 和 D. Michie 以及 N. J. Nillson 的利用搜索估价函数引导搜索的方法,P. E. Hart,Nillson 和 B. Raphael 的 A^* 算法,与或图上的启发式搜索 AO^* 算法以及各种**博弈树搜索**等。

启发式搜索的最大特点就是在搜索过程中使用与问题有关的启发信息来缩减搜索量,其一般过程

如下：

步骤 1　建立只含有初始节点 S 的搜索图 G，把 S 放入名为 OPEN 的未扩展节点表中；

步骤 2　建立扩展节点表 CLOSED，CLOSED 初始为空表；

步骤 3　若 OPEN 为空表，则搜索失败并退出；

步骤 4　把 OPEN 表上的第一个节点 node 移入 CLOSED 表；

步骤 5　若 node 为目标节点，则搜索成功并退出。问题的解就是图 G 中从 node 开始，沿着指针到达 S 的一条路径（节点的指针在步骤 7 建立）；

步骤 6　对节点 node 进行扩展，生成 node 的未在其祖先节点中出现过的后继节点集合 M，将 M 的成员添入图 G；

步骤 7　对 M 中的每一个未在 OPEN 表和 CLOSED 表中出现过的节点设置指针指向 node，并把这些节点放入 OPEN 表。对 M 中已在两表中的成员，确定是否需要更改其指向 node 的指针方向。对 M 中已在 CLOSED 表中的成员确定是否需要更改图 G 中通向它的每个后裔节点的指针方向；

步骤 8　按某种策略和规则或某种估价值重新排列 OPEN 表中的节点顺序；

步骤 9　返回到步骤 3。

上述过程中的步骤 8 对 OPEN 表上的节点进行排序，以便选出一个当前“最好”的节点在步骤 4 进行扩展。排序的策略和方法决定了搜索的性质和效率。

如果对 OPEN 表中的节点进行排序时不利用任何与问题有关的信息而取任意顺序或仅依据某种特定的简单策略和方式，则得到的过程就是无信息搜索或称为盲目搜索。典型的无信息搜索有宽度优先搜索和深度优先搜索等。一般来说，无信息搜索在搜索过程中要生成过多的节点，以至对许多现实问题而言很难或根本不能完成搜索任务。因而人们设法利用与问题有关的各种启发信息对 OPEN 表中的节点进行排序，从而使得搜索有可能沿着最有希望的路径区段进行。

要实现对节点的排序，关键是要给出某种测度，用以估算候选节点向目标节点逼近的“希望”程度。这种测度称为**搜索估价函数**。

估价函数通常记为 f，它提供了对节点的评价方法，能够确定有可能位于通向目标节点的最佳路径上的节点。可以根据问题从不同角度出发定义估价函数，例如节点处在最佳路径上的概率；节点与目标节点之间的距离测度或差异性测度等。估价函数选取的合适程度取决于所掌握的与问题有关的信息。

估价函数确定以后就可以根据节点的估价函数值在前述算法中的步骤 8 对节点重新排序。下面是在上述算法基础上给出的一个典型的利用估价函数的启发式搜索算法，其中假定所选估价函数 f 的函数值随节点的希望程度增大而减小。

步骤 1　把节点 S 放入 OPEN 表，计算 $f(S)$；

步骤 2　如果 OPEN 表为空，则失败退出，无解；

步骤 3　从OPEN 表中选择一个 f 值最小的节点 node_i。如果 f 值最小的节点有多个，且其中一个为目标节点时，选该节点为 node_i，否则选其中任一个节点作为 node_i；

步骤4　把 node_i 移出 OPEN 表，放入 CLOSED 表；

步骤 5　如果 node_i 是目标节点，则成功退出，求得一个解；

步骤 6　扩展 node_i，生成其全部后继节点。对于 node_i 的每个后继节点 node_j，进行以下工作：

（1）计算 $f(\text{node}_j)$；

（2）如果 node_j 不在 OPEN 表和 CLOSED 表中，则利用其估价函数 f 的值把它填入 OPEN 表并从 node_j 设一指向 node_i 的指针，以便找到目标节点后回溯解答路径；

（3）如果 node_j 已在 OPEN 表或 CLOSED 表中，则比较刚计算出的 $f(\text{node}_j)$ 和 node_j 已在表中的 f 值，若新的 f 值小，则以新值取代旧值；从 node_j 指向 node_i，而不是指向它的父节点；若 node_j 在 CLOSED 表中，则把它移回 OPEN 表；

步骤 7　转到步骤 2。

可以看出宽度优先和深度优先搜索等都是上述搜索算法的特例。比如选择 $f(\text{node}_i)$ 为 node_i 的深度即可得到宽度优先搜索。

估价函数的选择对启发式搜索的性能和结果有决定性的作用。搜索方法的优劣通常可以用渗透率和有效分支系数来衡量。渗透率定义为

$$P = \frac{L}{T}$$

式中，L 是所求得的到达目标节点的路径长度；

T 是搜索期间所生成的节点总数。

有效分支系数与 L 和 T 则有如下关系：

$$\frac{B(B^L - 1)}{B - 1} = T$$

B 值反映了搜索走向在目标节点方向上的集中程度。

启发式搜索可能是不完全的，它有可能得不到存在的最优解。从本质上说，这是牺牲完全性来换取高效率和满意解。启发式方法极大地提高了搜索效率，使得大量实际问题可以得到令人满意的解决，从而在许多领域，特别是在人工智能应用中取得了良好的效果，成为人工智能的主要技术之一。近年来人们在更为深刻和更为宽广的意义上提出了启发式程序设计的思想，从而把启发式搜索方法的研究提高到一个新水平。启发式方法的研究将继续对计算机科学和人工智能产生重大影响。

参考文献

1. 蔡自兴，等. 人工智能及其应用. 5版. 北京：清华大学出版社，2016

2. Mercadal P. Dictionary of artificial intelligence. New York：Van Nostrand Reinhold，1990

3. Shapiro S C. Encyclopedia of artificial intelligence，2nd ed. New York：John Wiley Sons Inc.，1992 （赵沁平）

qianzhao yitaiwang

千兆以太网（Gigabit Ethernet） 千兆技术仍然是以太技术，它采用了与10 M以太网相同的帧格式、帧结构、网络协议、全/半双工工作方式、流控模式以及布线系统。由于该技术不改变传统以太网的桌面应用、操作系统，因此可与10 M或100 M的以太网很好地配合工作。

千兆以太网技术有两个标准：IEEE 802.3z和IEEE 802.3ab。IEEE 802.3z制定了光纤和短程铜线连接方案的标准。IEEE 802.3ab制定了五类双绞线上较长距离连接方案的标准。

（1）IEEE 802.3z定义了基于光纤和短距离铜缆的1000Base-X，采用8B/10B编码技术，信道传输速度为1.25 Gbit/s，去耦后实现1000 Mbit/s传输速度。IEEE 802.3z具有下列千兆以太网标准：1000Base-SX只支持多模光纤，传输距离为220～550 m。1000Base-LX单模光纤，传输距离为5 km左右。1000Base-CX采用150 Ω屏蔽双绞线（STP），传输距离为25 m。

（2）IEEE 802.3ab定义基于5类UTP的1000Base-T标准，其目的是在5类UTP上以1000 Mb/s速率传输。 （金耀辉）

qianhou duanyan fangfa

前后断言方法（pre- and post-assertion method） 在语句前后分别加上前提条件（即前断言）和结果断言（即后断言），用程序设计逻辑证明程序正确性的方法。前提条件是该语句可执行的充分条件。结果断言指出语句执行中止后应具有的性质。前提条件又称前置断言，结果断言也称后置断言、后置条件等。

该证明方法源于R. Floyd，他在1967年提出在流程图的连线上加上断言使得程序到达该处时断言为真。后经C. A. R. Hoare发展成型，成为程序设计逻辑的主要部分。

前后断言的主要形式有两种：

（1）$Q\{S\}R$ 刻画语句S的部分正确性。意指：若前提条件Q满足，且S执行终止，则S终止时结果断言R成立。

（2）$\{Q\}S\{R\}$ 刻画语句S的完全正确性。意指：若前提条件Q满足，则S执行终止，且S终止时结果断言R成立。

将$Q\{S\}R$或$\{Q\}S\{R\}$的整体作为一个谓词公式，若该公式永真，则S相对于前提条件Q和结果断言R分别是部分正确或完全正确的。为证明复合语句$Q\{S_1;S_2\}R$的部分正确性，可在$S_1;S_2$之间插入断言Q'使得$Q\{S_1\}Q'$和$Q'\{S_2\}R$都永真。也可以插入多个断言Q_1,Q_2,Q_3等，使它们之间是蕴涵关系，用以反映证明思路。对其他语句构造可采用类似的方法。对于循环语句，除给出前后断言外，还须在循环中加入循环不变式，要求每次循环到达该处时不变式都成立。用前后断言可表示语句的公理语义。利用这些语义规则可帮助我们在程序正确性证明过程中确定该插入何种断言，以达到引导证明的目的（参见**最弱前置条件方法**）。前后断言方法，特别是结合最弱前置条件方法现已广泛用于规约语言及其支撑系统，如VDM，Larch等。

参考文献

1. Hoare C A R. An axiomatic basis for computer programming. CACM，1969，12(10)：576-580

2. Gries D. The science of programming. New York：Springer-Verlag，1981

3. Dijkstra E W. A discipline of programming. Englewood Cliffs，NJ：Printice Hall，1976 （伊波）

qianrushi caozuo xitong

嵌入式操作系统（embedded operating system） 为嵌入式电子设备提供的现代**操作系统**。嵌入的计算机及其操作系统与这些设备的其他部件密切地结合成一体。**嵌入式电子设备**泛指内部嵌有

计算机的各种灵巧电子设备，这些电子设备的应用范围涉及信息采集、信息交流、通信娱乐等多种应用领域。嵌入式操作系统是嵌入在这些设备内部的计算机操作系统，它为设备实现各种灵活功能提供信息处理系统平台。从技术发展来看，一旦诸如移动电话、掌上电脑、媒体视听等设备的内部嵌入了计算机，就为设备的多样服务功能提供了坚实的技术基础。

嵌入式操作系统的主要特点是，它需要满足多种多样嵌入式设备的功能需求和满足设备应用环境的需求，主要包括：①尽量节约设备的电池耗电，提供能源管理功能；②应用中有不同档次的实时性要求，特别是满足言语、视频影像等信息服务的及时性要求；③高可靠性要求，要防止信息丢失、泄漏、恶意破坏等；④操作系统的易移植性要求，满足在多种硬件环境下安装和配置的需要。

目前常见的嵌入式操作系统有 Windows CE、VxWorks 和从 Linux 发展出来的嵌入式 Linux；具有专门用途的掌上电脑的 Plam OS 以及用于移动电话的 Symbian 等。

为了提高系统的易移植性，嵌入式操作系统通常采用硬件抽象层（HAL）和板支持组件（BSP）的底层结构设计。

参考文献

1. 陈向群，杨芙清. 操作系统教程. 北京：北京大学出版社，2001

2. 尤晋元，史美林，等. Windows 操作系统原理. 北京：机械工业出版社，2001

3. Tanenbaum A S. Modern operating systems. 2nd ed. Prentice Hall，2001

（陈向群）

qianrushi jisuanji

嵌入式计算机（embedded computer）　作为一个信息处理部件，嵌入到应用系统中的**计算机**。

嵌入式计算机可以是**微型计算机**、**工作站**、**小型计算机**，甚至**巨型计算机**。但用得最多的还是**单片计算机**和单板计算机。一个应用系统可嵌入一台或多台计算机。由于嵌入式计算机埋藏在应用系统之中，一般不单独运行，甚至不暴露在现场操作人员的面前，所以在技术上具有如下特征：

（1）在功能和形体结构上都嵌入于应用系统之中，其体积、重量、外形尺寸、功耗等物理参数要适应所嵌入的应用系统所提供的条件，其性能参数要符合所嵌入的应用系统的需要。

（2）高可靠性　嵌入式计算机一般不进行日常检修，有的还运行于恶劣环境中，所以对可靠性要求很高，例如宇宙飞船和卫星上的嵌入式计算机要能经得起发射时的加速度、振动以及太空的恶劣工作环境。在元器件测试、机械结构设计、热设计、软件和硬件可靠性设计等方面要求十分严格。为提高可靠性，广泛使用了冗余技术，以保证部分硬、软件出故障后，系统整体仍能正确运行。

（3）自我检测功能　在应用系统工作时，往往不允许对嵌入式计算机进行维护。因此，在设计嵌入式计算机时，除了尽可能保证高可靠性外，还应考虑自我检测功能。要求嵌入式计算机能自动、迅速、准确地对故障进行检测、定位。有容错功能的还应根据故障情况对计算机系统进行重构。必要时，亦应便于对嵌入式计算机进行整体更换，以尽可能减少因嵌入式计算机故障而影响整个应用系统的正常工作。

（4）实时性　嵌入式计算机运行时有很强的实时性约束。嵌入式计算机大多从传感器接收输入信号，经处理后，其输出信号用于驱动各种执行机构，或显示出来供实时决策参考。

（5）通用机专用化　在一个特定的应用系统中，嵌入式计算机往往承担固定不变的专用任务。可以按应用系统的特定要求来设计专用计算机，但这是很不经济的。较好的办法是对通用计算机做适当的剪裁来满足不同应用系统的特定要求。

（6）软件固化　嵌入式计算机软件在某一特定的应用系统中往往固定不变，虽然可以修改和扩充，但一旦投入使用，将长时间不变，所以嵌入式计算机的软件常被固化。

（7）低能耗　应用系统对嵌入式计算机的供电通常有一定的限制，更有不少嵌入式计算机工作在依靠电池供电的环境中。因此，嵌入式计算机应满足低功耗的设计要求，除了在正常工作时尽可能节电外，还要根据不同的工作状态进行功耗管理，设置待机、休眠等不同模式，适时关闭某些部件或单元的电源。

发展简史

按照通用化、标准化、系列化的程度，嵌入式计算机的发展可分为 3 个阶段。

第一阶段（20 世纪 70 年代中至 80 年代初）为初始阶段，基本上属于军用标准阶段。在这个阶段，标准化工作主要着眼于指令系统结构等硬件措施上，虽然美国各军兵种所使用的嵌入式计算机开始

时曾有标准的军用指令系统结构,但各军兵种并未统一,形成了陆军的 NEBULA 结构;海军的 UYK-43,UYK-44,EMSP 以及 AYK-14 结构;空军的 MIL-STD-1750 A 结构。使用的编程语言也不统一,如陆军使用 NEBULA,空军使用 JOVIAL 和 PASCAL,海军使用 CMS-2 等。由于应用程序互不通用,影响了互相的交流。这个阶段的嵌入式实时操作系统一般都较简单,是为某种特定应用而设计的专用监控程序。直至 80 年代初才有了对应于某种微处理器芯片而设计的通用实时管理程序,如 Hunter & Ready Inc. 的 VRTX 和 Software Component Group Inc. 的 PSOS 等。它们主要用于以 Intel 8086 和 MC 68000 系列为中央处理器的嵌入式计算机系统,如美国空军的 MIL-STD-1750 A 嵌入式计算机采用了 VRTX 实时管理程序。

第二阶段为 80 年代中后期。在这个阶段,嵌入式计算机的通用化、标准化、系列化有了重大进展。从 1984 年起,美国国防部规定军内软件统一采用 Ada 语言编程,另外开发了 STARS 技术以补充 Ada 语言。所以在软件方面推行 Ada 语言和 STARS 技术,在硬件方面推行各种总线标准和接口标准是这个阶段的主要特征。这时嵌入式实时操作系统已趋完善,如 Real-time Performance 公司的 PRCore,Digital Research 公司的 Flex OS 等。军用嵌入式计算机除了使用军用标准计算机外,也使用民用计算机,以保持和现代新技术同步。另外,美国在推行 Ada 语言的基础上,提倡先写应用软件,然后根据应用的需要再来配置计算机系统的硬件和软件,从而保证嵌入式计算机硬、软件技术的先进性。

第三阶段为 80 年代末以后,嵌入式计算机的应用更加普及,市场迅速扩大。这个时期嵌入式计算机硬件方面的显著特征是采用高性能的 32 位 CISC(参见**复杂指令集计算机**)或 RISC(参见**精简指令集计算机**)处理器芯片,如 80386,80486,i960,SPARC,R2000,R3000,AMD 2900 等。而软件方面的特点是面向开放系统、多处理和集成开发工具,如 Software Component Group Inc. 推出了面向 32 位精简指令集计算机(RISC),复杂指令集计算机(CISC)处理器的嵌入式实时操作系统 PSOS + m,它保留了 PSOS 的特性,并具有多处理功能和丰富的开发工具,Intel 公司也针对其 32 位处理器,推出了嵌入式实时操作系统内核 iRMK Ⅲ 以支持 80386,80486 和 i960。

进入 21 世纪以后,超大规模集成电路工艺的发展,使嵌入式计算机进入了**片上系统**的时代。为了满足应用领域对高性能、小尺寸、低能耗的需求,往往将 32 位**微处理器**(CPU)、数字信号处理器(DSP)、专用加速部件、FLASH 存储器、内存、总线及外设接口等集成在单个芯片上,甚至还可以用系统级封装技术,把不同工艺条件下实现的数字、模拟、射频、传感器等部件包封在同一个管壳内,形成复杂的片上系统,以支撑许多实时应用系统。

分类与应用

嵌入式计算机的分类方法很多,按应用领域可分为军用计算机、工业用计算机和民用计算机。通常,军用计算机和工业用计算机的运行环境有特殊的要求(如对温度、湿度、振动、冲击、灰尘、腐蚀、辐射等),为了适应所嵌入系统的各种不同的恶劣环境条件,嵌入式计算机又可分为加固计算机、半加固计算机和未加固计算机,按照嵌入应用系统的计算机本身类型可以分为微型计算机、工作站、小型计算机、大型计算机等。嵌入式微型计算机又可分为单片计算机、单板计算机和多板计算机等。

计算机的嵌入式应用已成为计算机应用的主要形式之一,特别是单片计算机和片上系统已广泛用于家用电器(如电冰箱、自动洗衣机、照相机、空调机等),数码产品(如手机、数字电视机顶盒、数码摄像机、游戏机等),仪器仪表,医疗设备,数控机床,工业机器人,战略战术武器系统以及航天、测控和导航系统等。 (沈祖恩 唐志敏)

qianrushi xitong

嵌入式系统(embedded system) 以嵌入式处理器为核心部件的,用于执行独立功能的专用计算机系统。IEEE(国际电气和电子工程师协会)对嵌入式系统的定义是“用于控制、监视或者辅助装备、机器和设备运行的装置”。它由包括**微处理器**、定时器、微控制器、存储器、传感器、网卡等一系列微电子芯片与器件,和嵌入在存储器中的微型操作系统、控制应用软件组成,共同实现诸如实时控制、监视、管理、移动计算、数据处理等各种自动化处理任务。嵌入式系统以微电子技术、控制技术、计算机技术和通信技术为基础,强调硬件、软件的协同性与整合性,软件与硬件可剪裁性,以满足系统对功能、成本、体积和功耗等要求。嵌入式系统若以控制为目标,亦称**嵌入式控制系统**。

嵌入式系统已经有了近 30 年的发展历史。20 世纪 70 年代单片机的出现,使得汽车、家电、工业机

器、通信装置等各种产品可以通过内嵌电子装置来获得更佳的使用性能，如更易用、更快、更便宜。这些装置已经初步具备了嵌入式的应用特点，但是这时的应用只是使用 8 位的芯片，在唯一的**只读存储器**（ROM）中执行一些单线程的程序，无微型操作系统。

80 年代早期，嵌入式应用的程序员开始用商业级的“**操作系统**”编写嵌入式应用软件，以获取更短的开发周期，更低的开发费用和更高的开发效率，可以称得上是“嵌入式系统”。这个时期的操作系统是一个实时核，这个实时核包含了许多传统操作系统的特征，如任务管理、任务间通信、同步与相互排斥、中断支持、内存管理等功能。

90 年代以后，随着对实时性要求的提高，软件规模不断提升，实时核逐渐发展为实时多任务操作系统（RTOS），并作为一种软件平台逐步成为目前嵌入式系统的主流。复杂的嵌入式系统，例如个人数字助理（PDA）、手持电脑（HPC）等，具有与 PC 几乎一样的功能。许多公司看到了嵌入式系统的广阔发展前景，开始大力发展嵌入式操作系统，比较著名的有 Ready System 公司的 VRTX，Integrated System Incorporation（ISI）的 PSOS 和 IMG 的 VxWorks，QNX 公司的 QNX，Palm OS，WinCE，嵌入式 Linux，Lynx，Nucleux，以及国内的 Hopen，Delta Os 等嵌入式操作系统。

嵌入式系统的硬件构成

图 1 给出了用于测量和控制的嵌入式系统结构框图。

1. 嵌入式系统的核心部件

嵌入式系统的核心部件是各种类型的嵌入式处理器，相当于 PC 中的**中央处理器**（CPU）。按组成和功能又分为嵌入式微处理器（embedded microprocessor unit，EMPU），嵌入式微控制器（embedded microcontroller unit，EMCU），又称单片机，嵌入式数字信号处理器（embedded digital signal processor，EDSP）和嵌入式片上系统（system on chip，SOC）。

常见的嵌入式微处理器架构有 ARM、x86、MIPS、PowerPC、SH、Motorola 68K、ColdFire、SPARC 等，而 ARM、MIPS、PowerPC 是其中 3 种最主要的精简指令集计算机（RISC）架构。

2. 嵌入式系统的周边硬件

（1）嵌入式系统中必不可少的存储器　有 ROM（包括 EPROM，EEPROM），随机存取器（RAM）（包括静态随机存储器 SRAM，动态随机存储器 DRAM）和**快闪存储器**（flash memory）等。

（2）输入设备　一般包括**触屏**、按键、**键盘**、语音识别装置等。

（3）接口与总线　CPU 与外部设备接口通常指的是 I/O 接口，它包括并行接口和串行接口。串口的典型代表是 RS-232-C、RS-485 和 USB，以及红外接口等。嵌入式系统中各部分之间的数据传送也有各种总线，其中主要有：ISA 总线、PCI 总线、I^2C 总线、SPI 总线、PC104 总线、I^2S 总线等。

3. 嵌入式系统开发平台

嵌入式系统在开发阶段，需要有硬件平台的支持。通常嵌入式处理器的芯片厂商会提供评估版，但由于其配置有限，往往不能满足应用开发的需要，所以又有一些公司推出了一些较通用的开发平台，目前应用最多的是 ARM 平台。

嵌入式操作系统及开发环境的构建

嵌入式操作系统（embedded operation system，EOS）是一种用途广泛的系统软件，负责嵌入系统的全部软、硬件资源的分配、任务调度，控制、协调并

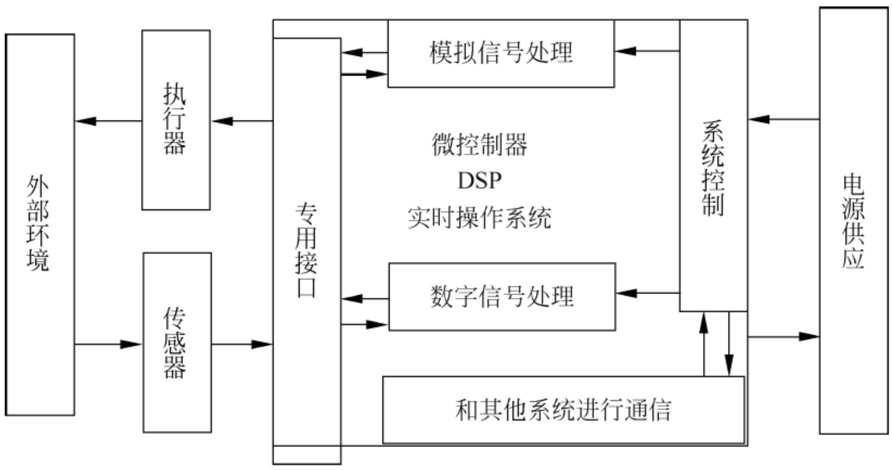


图 1　用于测量和控制的嵌入式系统

发活动。

嵌入式操作系统最典型的代表是嵌入式 Linux。它以开源、可定制和强大的网络功能为嵌入式系统的网络互联和功能扩展提供了必备条件和广阔的发展空间。此外 VxWorks、Windows CE 以其高可靠性、实时性,有着广泛的应用领域。

在嵌入式系统开发中,通常目标平台资源有限,所以需要在宿主机(通常为 PC)上进行编程、编译和调试,以生成针对目标平台的代码,这个过程称为交叉编译。交叉编译环境的建立需要以下步骤:

(1)确定并获得交叉编译工具链。

(2)内核移植 针对硬件平台,在现有内核源码的基础上,添加和修改相应代码,进行合理配置,然后进行交叉编译,生成内核映像文件,下载到目标平台。

(3)制作根文件系统 系统启动和随后的基本操作必须建立根文件系统,它包含链接库、内核模块、内核映像、典型设备文件和主要的系统应用程序等。

嵌入式系统涉及计算机、操作系统、信息处理、网络通信、集成制造等多种学科,这些领域的进步,将促进嵌入式系统的更新换代。随着时代发展的需求和技术的进步,嵌入式系统的内涵和外延不断扩大。嵌入式系统正逐渐向网络化、信息化方向发展。物联网、信息物理融合系统,CPS(Cyber Physics System),已成为信息科学领域的研究热点,而嵌入式系统正是实现物联网和 CPS 的有效途径。具备传感、计算、通信、存储能力的嵌入式系统,将是未来一段时间的发展方向。它已经和正在使传统的控制方式和手段发生重大的变化。

参考文献

1. 慕春棣. 嵌入式系统的构建. 北京:清华大学出版社,2004

2. 邹思轶. 嵌入式 Linux 设计与应用. 北京:清华大学出版社,2002

3. Karim Yagbmour. 构建嵌入式 Linux 系统. O'Relly Taiwan 公司,译. 北京:中国电力出版社,2004

(慕春棣)

qianghua xuexi

强化学习(reinforcement learning) 研究在动态环境中,通过环境的奖惩反馈,学习环境状态到行为映射的策略,以使从环境中获得的累积奖赏总和最大的**机器学习**理论、模型与方法。例如,学习骑自行车就可以看作是一个强化学习问题。强化学习问题中,并不需要事先提供正例和反例的样本集合,而是通过试错(trial-and-error)的方法来发现最优行为策略。强化学习方法结合统计技术和动态规划方法,通过估计在某一环境状态的效用函数,从而确定最优行为策略。强化学习技术起源于**人工智能**、动物学习和自动控制,在计算机、控制、心理学、经济学等领域均有研究和应用。

强化学习模型和基本概念

强化学习方法常用于解决顺序型决策任务,并可以通过马尔可夫决策过程(Markov Decision Process,MDP)模型建模。基本的强化学习模型包含:环境状态集合 S,系统行为集合 A,状态转移函数 P,瞬时奖赏函数 R。如学习骑自行车的问题中,“自行车静止”“自行车行驶”“自行车摔倒”等即属于环境状态集合;而“上车”“加速”“刹车”“下车”“转弯”等属于系统行为集合;“撞人”“顺利到达”等则属于奖赏集合。如果速度较快且转弯较急的情况,则有一定的概率会使自行车摔倒。这说明在某些系统行为的作用下,环境状态会以一定概率向其他状态发生迁移。

学习系统在开始学习前,如果状态转移函数 P 和瞬时奖赏函数 R 已知,则称为模型已知。此时顺序型决策任务可以采用动态规划技术求解最优策略。所谓模型未知,是指学习系统在开始学习前状态转移函数 P 和瞬时奖赏函数 R 未知。此时可以采用强化学习技术求解最优策略。策略是指状态到行为的映射,常见的包括确定性策略、随机性策略、最优策略等。确定性策略是指该映射是确定性的;而随机性策略是指该映射是不确定性的,是定义在行为集合上的概率函数。最优策略是指不存在任何一种其他策略(包括确定性策略和随机性策略),所从环境中获得的累积奖赏和能超过最优策略所获得的。

强化学习系统所面临的环境是由环境模型所确定。但由于模型中状态转移函数 P 和瞬时奖赏函数 R 未知,学习系统只能够依赖每次试错所获得的瞬时奖赏来选择策略。考虑到环境模型的不确定性和目标的长远性,强化学习方法在策略和瞬时奖赏之间构造“值函数”(即状态的效用函数),用于策略的选择和策略的评估。在某个策略 π 指导下,状态 s 的值函数等于从 s 状态往后所获得的累积奖赏和。例如,骑自行车去目的地,目标是“更快”“更好”地达到目的地,时间要尽可能短,而且不能“摔倒”“撞

人”。因此,骑车者需要选择在某某状态下“加速”某某状态下“刹车”、某某状态下“转弯”等,这就属于某个策略 π。如果骑车者采用这个策略,从某状态往后所获得的累积奖赏和,即称为该状态在此策略下的“值函数”。显然,使每个状态值函数最大的策略即是系统的“最优策略”。

强化学习算法

基本的强化学习算法结合了动态规划方法和蒙特卡罗方法。在动态规划技术中,在环境模型已知前提下,从任意设定的策略出发,可以采用策略迭代或值迭代逼近最优的值函数和策略。举例来说,骑车者起初采用慢速转弯的策略,但如果发现快速转弯可以获得更大的值函数时,骑车者就采用策略迭代(改用快速转弯策略)去逼近最优的值函数和策略。

根据 Bellman 公式,值函数可以采用当前状态下的瞬时奖赏和下一状态值函数的递归定义形式。由于强化学习问题中模型未知,学习系统无法直接使用动态规划技术进行值函数计算。因此在实际中采用采样的方法进行值函数估计,其中最主要的方法之一是蒙特卡洛采样。

经典的强化学习算法有 Q 学习算法和 Sarsa 算法。强化学习算法主要包括两个步骤,一是根据当前的值函数,进行当前状态的动作选择;二是根据在线获得的瞬时奖赏,进行当前状态的值函数更新。在动作选择时,需要平衡学习系统的搜索(exploration)和利用(exploitation)。通常会在学习的初期,加大搜索,而在学习后期,强调利用。强化学习算法中常用的动作选择策略有 ε-greedy 策略和 Boltzmann 策略等。

值函数更新也有两种主要的方法,一种称为在策略(on-policy)更新,另一种称为离策略(off-policy)更新。在策略更新中,值函数采用当前策略所选择的下一状态的值函数进行更新。而在离策略更新中,值函数采用当前值函数所确定的最优策略下的状态值函数进行更新。Q 学习算法采用离策略更新方法,Sarsa 算法采用在策略更新方法。

强化学习发展

在大规模顺序决策任务和不满足马尔可夫性的顺序决策任务中,经典的强化学习算法收敛速度慢甚至失效。因此,现代强化学习技术主要解决以上顺序决策任务中的最优策略学习。近年来主要的发展包括有:

强化学习函数估计　采用参数化的函数来拟合值函数,以解决大规模和连续状态下强化学习的“维度灾难”问题。如在骑自行车问题中,路面和自行车状态无法离散化,则可以采用参数化的函数来拟合值函数。

关系强化学习　采用逻辑马尔可夫模型来建模,侧重于表达状态和行为之间的关系,实现对强化学习问题的抽象。特别是许多先验知识(如停车前需要刹车减速)很容易以关系的方式表达,并结合进关系强化学习技术中。

分层强化学习　通过将一个复杂任务分解为若干简单的子任务的集合,实现不同层次的抽象,以加速在大规模任务中的学习。如骑自行车问题,可以分解为直道行驶、弯道行驶、爬坡行驶、下坡行驶等多个子任务,学习者可以分别学习不同子任务的最优策略。

部分感知强化学习　当学习系统的观察与实际状态不一致时,采用部分感知马尔可夫决策模型建模,通过部分感知强化学习学习最优策略。部分感知强化学习中往往会应用到强化学习函数估计技术。在骑自行车问题中,学习系统很难准确得到行驶速度、转弯角度、距离等具体的度量,对环境状态的感知存在偏差。因此,在部分感知强化学习技术中,往往需要历史记忆或者不确定性策略等方式,学习最优策略。

多 agent 强化学习　当环境同时存在多个强化学习系统时,强化学习系统不仅要考虑环境本身的动态性,还要考虑其他学习系统导致的不确定性。多 agent 强化学习需要考虑 agent 之间的关系。如在道路上同时存在多个骑自行车者,如何协调不同骑车者之间的行为,以获得多个骑车者的最优策略,即属于多 agent 强化学习问题。

强化学习迁移　使用在其他学习任务中学习的经验,来加速在新任务中的学习。主要包括有知识迁移和行为迁移等。如将骑三轮童车的经验,帮助学习骑自行车,就属于强化学习迁移问题。

参考文献

1. Sutton R S, Barto A G. Reinforcement learning: An introduction. MIT Press, 1998

2. 高阳,陈世福,陆鑫. 强化学习研究综述. 自动化学报, 2004, 30(1): 86-100

(高阳)

Qiaomusiji cengci

乔姆斯基层次(Chomsky hierarchy)　由 N. Chomsky 提出的形式语言的分层(参见**形式语言理**

论)。它研究四类文法,即**短语结构文法**、**上下文有关文法**、**上下文无关文法**和**正则文法**之间存在的层次关系,以及由该四类文法产生的语言族所形成的层次。

短语结构文法 一个四元组 $G=(\Sigma,V,S,P)$,其中 V 是非终结符的有穷字母表,Σ 是终结符的有限字母表,P 是生成式的有限非空集,P 中的生成式均为 $\alpha\to\beta$ 的形式,$\alpha\in(\Sigma\cup V)^+$,$\beta\in(\Sigma\cup V)^*$,$S(\in V)$称为开始符号。短语结构文法是一种非受限文法,也称为0型文法和半丘系统。对短语结构文法中的生成式作某些限制,即得到上下文有关文法、上下文无关文法和正则文法。

上下文有关文法 一种特殊的短语结构文法,也称为1型文法。它限制 P 中的生成式均为 $\alpha_1A\alpha_2\to\alpha_1\beta\alpha_2$ 的形式,其中 $A\in V$,$\alpha_1,\alpha_2,\beta\in(\Sigma\cup V)^*$,$\beta\neq\varepsilon$。对这类文法,在派生中只有"上下文"$\alpha_1,\alpha_2$ 分别出现在 A 的左、右邻近时,才允许用 β 替换 A,故云"上下文有关"。

上下文无关文法 一类特殊的上下文有关文法,也称为2型文法。它限制 P 中的生成式均为 $A\to\beta$ 的形式,其中 $A\in V$,$\beta\in(\Sigma\cup V)^*$,$\beta\neq\varepsilon$。

正则文法 一类特殊的上下文无关文法,也称为3型文法。这类文法要求 P 中的生成式均为 $A\to aB$ 或 $A\to a$ 的形式,其中 $A,B\in V$,$a\in\Sigma$。在有些文献中也称左线性文法或右线性文法为正则文法。它们都是和正则文法等价的。称文法是左(右)线性文法,如果 P 中的生成式都是 $A\to Bw(A\to wB)$ 或 $A\to w$ 的形式,其中 $A,B\in V$,$w\in\Sigma^*$。

由 i 型文法 $G=(\Sigma,V,S,P)$ 产生的语言称为 i 型语言,$i=0,1,2,3$,记作 $L(G)$,即

$$L(G)=\left\{x\in\Sigma^*\mid S\overset{*}{\underset{G}{\Rightarrow}}x\right\}$$

从四类文法的定义可见,它们之间存在明显的层次关系,从而,四个语言类也存在层次关系。若令 $\mathscr{L}_i$ 表示 i 型语言类,$i=0,1,2,3$,则 $\mathscr{L}_3\subseteq\mathscr{L}_2\subseteq\mathscr{L}_1\subseteq\mathscr{L}_0$。由于 $L=\{a^nb^n\mid n\geqslant1\}\in\mathscr{L}_2$,但不是3型语言,故 $\mathscr{L}_3$ 是 $\mathscr{L}_2$ 的真子集;又由于 $L=\{a^{2^i}\mid i\geqslant1\}\in\mathscr{L}_1$,但不是2型语言,所以 $\mathscr{L}_2$ 是 $\mathscr{L}_1$ 的真子集。类似地,在 $\mathscr{L}_0$ 中也存在不属于 $\mathscr{L}_1$ 的语言,故 $\mathscr{L}_1$ 也是 $\mathscr{L}_0$ 的真子集。这就是所谓的乔姆斯基层次。在有些文献中也称为乔姆斯基分层或乔姆斯基谱系。

关于语言的描述,除了从生成的角度,即文法的角度之外,还可从识别的角度,即自动机的角度来描述。相应于四类文法有四类自动机,即图灵机、线性有界自动机、下推自动机和有限自动机。并证明了:图灵机识别的语言类恰为 $\mathscr{L}_0$;线性有界自动机识别的语言类恰为 $\mathscr{L}_1$;下推自动机识别的语言类恰为 $\mathscr{L}_2$;有限自动机识别的语言类恰为 $\mathscr{L}_3$。有限状态自动机和正则文法的等价性是由 N. Chomsky 和 G. A. Miller 给出的,N. Chomsky 和 J. Evey 证明了下推自动机和上下文无关文法的等价性。S. Y. Kuroda证明了线性有界自动机与上下文有关文法的等价性。N. Chomsky 证明了图灵机与短语结构文法的等价性。

参考文献

Hopcroft J E, Ullman J D. Introduction to automata theory, languages, and computation. Boston, MA: Addison-Wesley,1979

(苏锦祥)

Qiaomusiji fanshi

乔姆斯基范式(Chomsky normal form) 一种由 N. Chomsky 提出的**上下文无关文法(2型文法)**的规范化形式。它对上下文无关文法生成式的形式加以某种限制。乔姆斯基证明了任意的上下文无关文法都存在等价的乔姆斯基范式。

若上下文无关文法 $G=(\Sigma,V,S,P)$ 中的生成式均为 $A\to BC$ 或 $A\to a$ 的形式,则称 G 为具有乔姆斯基范式的文法,其中 $A,B,C\in V$,$a\in\Sigma$,缩写为 CNF。N. Chomsky 在1959年证明了任何不含 ε 的上下文无关语言都可由乔姆斯基范式产生,这就是著名的乔姆斯基范式定理。实际上,乔姆斯基证明了更强的结果,即证明了:每个不含 ε 的上下文无关语言,均可由一个如下所述的上下文无关文法 $G=(\Sigma,V,S,P)$ 产生,其中所有的生成式均为 $A\to BC$ 或 $A\to a$ 的形式,$A,B,C\in V$,$a\in\Sigma$,$B\neq C$且若 $A\to\alpha_1B\alpha_2$,$A\to\beta_1B\beta_2$ 是 P 中的生成式,则 $\alpha_1=\beta_1=\varepsilon$ 或 $\alpha_2=\beta_2=\varepsilon$,$\alpha_1,\alpha_2,\beta_1,\beta_2\in(\Sigma\cup V)^*$。

根据乔姆斯基范式定理还可构造出一个等价的乔姆斯基范式,这种范式不仅使相应的上下文无关文法的生成式异常简单而且规范。这对研究相应的上下文无关语言的结构很有意义。例如,由乔姆斯基范式所产生的任意串对应的推导树都是一棵二叉树。

参考文献

Hopcroft J E, Ullman J D. Introduction to automata theory, languages, and computation. Boston, MA: Addison-Wesley, 1979

(苏锦祥)

Qiebixuefu bijin

切比雪夫逼近（Chebyshev approximation）

又称最佳一致逼近（参见**数值逼近**）。

令度量 $p=\infty$，且 $H=H_n$ 为次数不大于 n 的多项式函数集合。设 $f\in C[a,b]$，若 $P\in H_n$ 满足

$$E_n(f)\equiv\inf_{Q\in H_n}\|f-Q\|_\infty=\|f-P\|_\infty$$

则称 P 为 f 的次数不大于 n 的切比雪夫逼近多项式函数，称 $E_n(f)$ 为 H_n 对给定函数 f 的最小偏差。且有

$$E_0(f)\geqslant E_1(f)\geqslant\cdots,\quad \lim_{n\to\infty}E_n(f)=0$$

事实上，这样的多项式 P 是存在且唯一的。

切比雪夫定理：H_n 中的多项式函数 P 成为 $C[a,b]$ 中给定函数 f 的切比雪夫逼近多项式函数的充要条件是在 $[a,b]$ 上存在一组分点（称为偏差点组或交错点组）

$$a\leqslant x_1<x_2<\cdots<x_m\leqslant b,\quad m\geqslant n+2$$

使得

$$f(x_i)-P(x_i)=(-1)^i\lambda$$

$$|\lambda|=\|f-P\|_\infty,\quad i=1,2,\cdots,m$$

成立。

由定理知，当 $n=0$ 时，零次最佳一致逼近多项式为

$$P(x)=\frac{m+M}{2}$$

其中

$$m=\min_{a\leqslant x\leqslant b}f(x),\quad M=\max_{a\leqslant x\leqslant b}f(x)$$

当 $n=1$ 时，若 f 在 $[a,b]$ 上的二阶导数 f'' 连续且不变号，则一次最佳一致逼近多项式为

$$P(x)=a_0+a_1x$$

其中

$$a_1=\frac{f(b)-f(a)}{b-a},$$

$$a_0=\frac{f(a)+f(x_1)}{2}-a_1\frac{a+x_1}{2}$$

式中，x_1 是方程 $f'(x)=a_1$ 的根。

当 $n\geqslant 2$ 时，可由定理得到的列梅兹算法来逐次逼近求得。但实际计算很困难，一般很少使用。往往借助切比雪夫多项式 $T_n(x)=\cos(n\arccos x)$ 的性质来求近似的最佳一致逼近多项式。

使用三角多项式来逼近 $C_{2\pi}$ 中的函数时，可得到完全平行的结果。

参考文献

Davis P J. Interpolation approximation. Waltham, MA：Blaisdell, 1963

（沈毅）

qingjing yansuan

情景演算（situation calculus）　一种将情景作为受量词约束的一阶对象，对变化进行描述和推理的逻辑。其基本概念最早由 John McCarthy 在 1963 年针对动态问题域中**知识表示**和推理而提出。表示方法以**一阶谓词逻辑为主**。

情景演算中使用情景、动作和流三种基本组成元素。动作表示能够引起世界改变的单个瞬时事件。对情景有多种不同的理解方式，J. McCarthy 和 P. Pat Hayes 将情景视作“在某一瞬时的完整的世界状态”。动作的发生引起情景的变化，而 Raymond Reiter 将其等同于“世界中所发生的动作的历史”，即从初始情景 S_0 开始执行的有限动作序列。不论在哪种理解方式中，情景都是一个受量词约束的一阶逻辑对象，对变化具有很强的表达能力。流则表示世界可能随情景而变的某种属性，通常用来描述动作的效果。

在表达形式上，动作是以论域中某些对象为参数的项，比如可以用 $put(x,y)$ 表示将对象 x 放到对象 y 的上面；对于情景，一个形如 $do(a,s)$ 的谓词项，表示主体在情景 s 下执行动作 a 所导致的情景，按照 R. Reiter 的理解方式，$do(a,s)=do(a',s')$ 当且仅当 $a=a'$、$s=s'$；而流则是值随情景变化的谓词或函数，其最后一个参数为情景项，其余参数可以为动作或问题域中的对象。流分为关系流和函数流两种，前者只有真和假两个可能值，而后者则可以为一定范围内的值。比如：关系流 $holding(x,s)$ 表示在情景 s 下主体正持有对象 x，函数流 $location(s)$ 表示在情景 s 下，主体所处的空间位置，专有的关系流 $Poss(a,s)$ 表示在情景 s 下动作 a 是否可被执行。

作为一种基于逻辑的时序推理形式系统，情景演算为动态系统中的问题求解提供了一种有效框架，是针对动作和变化进行推理的一种强有力的方法，在规划求解、逻辑程序设计等**人工智能**领域有着重要应用。

参考文献

1. 史忠植. 高级人工智能. 北京：科技出版社，1998

2. Fangzhen Lin. Situation calculus. In：Van Harmelen F, Lifschitz V and Porter B, eds. Handbook of Knowledge Representation, Elsevier, 2008

3. Levesque H, Pirri F, Reiter R. Foundations

for the situation calculus. Electronic Transactions on Artificial Intelligence, 1998, 2(3-4): 159-178

（欧阳丹彤　崔仙姬）

qumian

曲面(surface)　**曲线**上所有点运动产生的轨迹。曲面一般分为解析曲面和不能用初等解析式表示的自由型曲面。在**计算机图形学**中,曲面是表示和构造复杂外形的重要数学工具。

解析曲面的显式或隐式表示历史悠久,而适合计算机处理的**参数曲面**则在近几十年内得到迅速发展。1963 年 Ferguson 用 4 个角点的位置矢量及其 2 个方向切矢量定义三次曲面。1964 年 Coons 用封闭的 4 条边界曲线定义一个曲面片。同年,Schoenberg 提出了参数样条曲面的形式。1971 年,Bézier 提出了一种用控制多边形定义曲面的方法。1972 年,de Boor 给出了 B 样条的标准计算方法。1974 年,Gordon 和 Riesenfeld 将 B 样条理论用于形状描述,提出 B 样条曲面。20 世纪 80 年代后期,Piegl 和 Tiller 将有理 B 样条发展成非均匀有理 B 样条(NURBS)。当前,它已成为产品几何外形描述的工业标准。用 NURBS 曲面可统一表示初等解析曲面、有理与非有理 Bézier 以及非有理 B 样条曲面。

对于曲面,主要研究它的数学表示、连续性及其在实际中的应用,重点研究用计算机构造具有一定连续性的实用曲面的理论与算法。

曲面的连续性指的是曲面片与曲面片之间的连接光滑程度。如果两曲面具有公共连接线,称它们是位置连续的或是 C^0 的,即具有零阶参数连续性。一般的曲面参数连续性即 C^n 连续性定义为:当且仅当两曲面 $p(s,t)$ 与 $q(u,v)$ 沿它们的公共连接线 $p(\gamma)=q(\gamma)$ 处处具有直到 n 阶的连续偏导矢,则称它们沿该连接线具有 n 阶参数连续性(即 C^n 连续性)或是 C^n 的。即

$$\frac{\partial^{i+j}}{\partial_s^i\partial_t^j}p(\gamma)=\frac{\partial^{i+j}}{\partial_u^i\partial_v^j}q(\gamma),\quad i+j=1,2,\cdots,n$$

两参数曲面的零阶几何连续性即 G^0 连续性是与 C^0 连续性一致的。

两参数曲面的 G^1 连续性又称为切平面连续性,其定义为:两曲面沿它们的公共连接线具有 G^1 连续性或是 G^1 的,当且仅当它们沿该公共连接线处处具有公共的切平面或公共的曲面法线。

两参数曲面的 G^2 连续性又称为曲率连续性,其定义为:两曲面沿它们的公共连接线具有 G^2 连续性或是 G^2 的,当且仅当它们沿该公共连接线处处具有公共的切平面外,又具有公共的主曲率,并在两个主曲率不相等时具有公共的主方向,或一致的杜潘标线。

一般的曲面几何连续性即 G^n 连续性定义为:两曲面 $p(s,t)$ 与 $q(u,v)$ 沿它们的正则公共连接线具有 G^n 连续性或是 G^n 的,当且仅当其中之一,譬如 q,可被重新参数化为 $\bar{q}(\bar{u},\bar{v})$,以至于它们沿该公共连接线是 C^n 的。即

$$\frac{\partial^{i+j}}{\partial_s^i\partial_t^j}p(\gamma)=\frac{\partial^{i+j}}{\partial_{\bar{u}}^i\partial_{\bar{v}}^j}\bar{q}(\gamma),\quad i+j=1,2,\cdots,n$$

在曲面造型中(参见**几何造型**),一般只用到 C^1, C^2 和 G^1, G^2 连续。

参考文献

1. Farin G. Curves and Surfaces for Computer Aided Geometric Design. Academic Press Inc., 1993

2. Mortenson M E. Geometric Modeling. John Wiley & Sons, 1985

3. 孙家广,等. 计算机图形学. 3 版. 北京: 清华大学出版社,1998

（孙家广　冯结青）

quxian

曲线(curve)　一个点在空间运动的轨迹。曲线一般分为解析曲线和不能用初等解析式表示的自由型曲线。在**计算机图形学**中,曲线是表示和构造复杂外形的重要数学工具和基础。

解析曲线的显式或隐式表示历史悠久,而适合计算机处理的**参数曲线**则在近几十年内得到迅速发展。1963 年 Ferguson 用三次参数法构造曲线。同年,Schoenberg 提出了参数样条曲线的形式。1971 年 Bézier 提出了一种用控制多边形定义曲线的方法。1972 年,de Boor 给出了 B 样条的标准计算方法。1974 年,Gordon 和 Riesenfeld 将 B 样条理论用于形状描述,提出 B 样条曲线。20 世纪 80 年代后期,Piegl 和 Tiller 将有理 B 样条发展成非均匀有理 B 样条(NURBS)。当前,它已成为描述曲线的主流方法,用 NURBS 可统一表示初等解析曲线、有理与非有理 Bézier 以及非有理 B 样条曲线。

对于曲线,主要研究它的数学表示、连续性及其在实际中的应用,重点研究用计算机构造具有一定连续性的实用曲线的理论与算法。

曲线的连续性指的是曲线段与曲线段之间的连接光滑程度。由调和函数构造的参数曲线,其自身(即在参数 t 的取值区间内)的连续性是由调和函数决定的。而对于几条参数曲线段首尾相接构成的参

数曲线，如何保证连接处具有合乎要求的连续性是关键问题。图 1 中给出 4 条参数曲线段 $Q_1(t)$ 和 $Q_2(t)$，$Q_3(t)$ 和 $Q_4(t)$，$t\in[0,1]$：

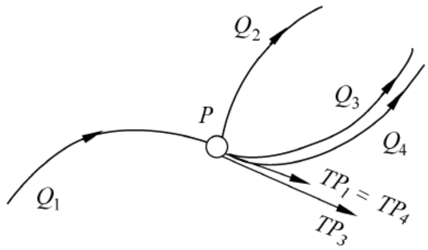


图 1　曲线段之间的连续性

（1）若 $Q_1(1)$ 和 $Q_2(0)$ 的端点重合于 P，则 $Q_1(t)$ 和 $Q_2(t)$ 在 P 点处有 C^0 和 G^0 连续。

（2）若 $Q_1(1)$ 和 $Q_3(0)$ 在 P 点处重合，且其在 P 点处的切矢量方向相同，大小不等，则 $Q_1(t)$ 和 $Q_3(t)$ 在 P 点处有 G^1 连续。

（3）若 $Q_1(1)$ 和 $Q_4(0)$ 在 P 点处重合，且其在 P 点处的切矢量方向相同，大小相等，则 $Q_1(t)$ 和 $Q_4(t)$ 在 P 点处有 C^1 连续。

（4）若 $Q_1(1)$ 和 $Q_4(0)$ 在 P 点处已有 C^0，C^1 连续且其 $Q''_1(1)$ 和 $Q''_4(0)$ 的大小和方向均相同，则 $Q_1(t)$ 和 $Q_4(t)$ 在 P 点处有 C^2 连续。推而广之，若 $Q_1^n(1)$ 和 $Q_4^n(0)$ 在 P 点处的大小和方向均相同，则说 $Q_1(t)$ 和 $Q_4(t)$ 在 P 点处有 C^n 连续。

（5）若 $Q_1(t)$ 和 $Q_3(t)$ 在 P 点处已具有 G^0，G^1 连续，且其 $Q''_1(1)$ 和 $Q''_3(0)$ 的方向相同，但大小不相等，则说 $Q_1(t)$ 和 $Q_3(t)$ 在 P 点处有 G^2 连续。

在曲线造型中（参见**几何造型**），一般只用到 C^1，C^2 和 G^1，G^2 连续。在实际应用中，应适当选择曲线段的连续性，使造型物体既能保证其光滑性，也能保证其美观性。

参考文献

1. Farin G. Curves and Surfaces for Computer Aided Geometric Design. Academic Press Inc., 1993

2. Mortenson M E. Geometric Modeling. John Wiley & Sons, 1985

3. 孙家广，等. 计算机图形学. 3 版. 北京：清华大学出版社，1998

（孙家广　冯结青）

quanpindai shuzi yinpin de bianma

全频带数字音频的编码（coding of full band digital audio）　对全频带数字声音信号进行数据压缩和编码处理的方法与技术。声音由许多不同频率的谐波组成，谐波的频率范围称为声音信号的带宽。人耳可听见的各种声音，如音乐声、风雨声、汽车声等，其带宽为 20～20 kHz，它们称为全频带音频信号。全频带音频信号数字化之后的数据量很大。以 CD 盘片上所存储的立体声高保真数字音乐（采用 PCM 编码）为例，1 小时的数据量大约是 635 MB。为了降低存储成本和提高通信效率（降低传输带宽），对全频带数字音频进行数据压缩是十分必要的。

全频带数字音频的数据压缩也是完全可能的，其依据是声音信号中包含有大量的冗余信息，再加上还可以利用人的听觉感知特性，因而产生了许多压缩算法。一个好的数据压缩算法应做到压缩比高，声音失真小，算法简单，编码器和解码器的成本低。

全频带数字音频的压缩编码技术可以分成 2 类：无损压缩和有损压缩。无损压缩主要是依据声音波形本身的信息相关性，通过消除冗余来进行数据压缩，它不会破坏原来的信息，播放时重建的声音与原始声音完全相同，但数据压缩比不高。有损压缩是在丢失一部分信息的情况下，按用户指定的压缩比（或比特率），尽可能保持接近原来的音质，所以数据压缩比比较高。

全频带音频的无损压缩编码方法有多种，目前使用比较普遍的有：Monkey's Audio（文件格式为 APE）、FLAC、ALAC（文件格式为 M4A）等，它们能够在精确再现原音频信号的前提下将数据量压缩到尽可能小，通常可压缩 50% 左右的数据量。主要应用于高保真音频作品的发布、播放和储存等场合。

全频带音频的有损压缩编码大多使用所谓的“心理声学模型”来达到大幅度压缩数据的目的，所以这种压缩编码方法也称为感知音频编码。图 1 是感知音频编码的基本原理框图。编码过程一般分为 3 个阶段，第一阶段通过时间/频率变换和心理声学分析，揭示原始声音中与人耳感知无关的信息，然后在第二阶段通过量化和编码予以抑制，第三阶段再使用熵编码消除声音信息中的统计冗余。

表 1 是几种常用的全频带音频的压缩编码技术和标准。其中 MPEG-1 音频压缩编码是国际上第 1 个全频带音频数据压缩的国际标准，它分为三个层次：层 1 的编码较简单，主要用于数字盒式录音磁带；层 2 的算法复杂度中等，其应用包括数字音频广播（DAB）和 VCD 等；层 3 的编码较复杂，主要用于因特网上高质量声音的传输。现在流行的 MP3 音

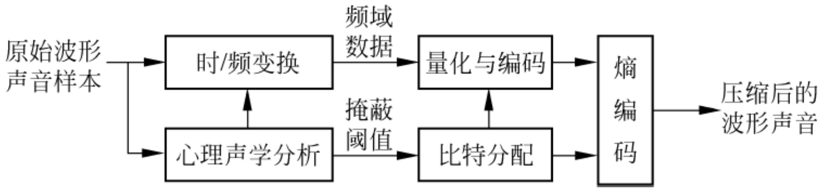


图 1　感知音频编码的原理框图

表 1　几种常用的全频带音频压缩编码技术和标准

文件类型	编码类型	效果	主要应用	开发者
WAV	未压缩	声音达到 CD 品质	支持多种采样频率和量化位数,获广泛支持	微软公司
FLAC	无损压缩	压缩比为 2:1 左右	高品质数字音乐	Xiph. Org 基金会
APE	无损压缩	压缩比为 2:1 左右	高品质数字音乐	Matthew T. Ashland
M4A	无损压缩	压缩比为 2:1 左右	QuickTime,iTunes,iPod,Real Player	苹果公司
MPEG-1 Audio 层 3(MP3)	有损压缩	压缩比为 8:1～12:1	互联网,MP3 音乐	ISO
WMA	有损压缩	压缩比高于 MP3,使用数字版权保护	互联网,音乐	微软公司
AC3	有损压缩	压缩比可调,支持 5.1、7.1 声道	DVD,数字电视,家庭影院等	美国 Dolby 公司
MPEG-2(AAC)	有损压缩	支持 5.1、7.1 声道	DVD,数字电视,家庭影院等	ISO

乐就是采用 MPEG-1 层 3 编码的数字音乐，它以 10 倍左右的压缩比降低了数字音频的数据量,一张普通 CD 光盘上可以存储大约 100 首 MP3 歌曲。

MPEG-2 有 2 种音频压缩编码。第 1 种(称为 MPEG-2 audio)采用与 MPEG-1 相同的方案,层 1,层 2 和层 3 的结构也相同,但它能支持 5.1 声道和 7.1 声道的环绕立体声。第 2 种(称为 MPEG-2 AAC)采用更先进的技术,效果也更好。

杜比数字 AC-3(Dolby Digital AC-3)是美国杜比公司开发的多声道全频带音频编码系统,目前在数字电视、DVD 和家庭影院中广泛使用。

WMA(Windows Media Audio)是微软公司开发的一种数字音频压缩技术,苹果公司的 iTunes 也提供对它的支持。一般而言,音质相同的情况下 WMA 音频比 MP3 音频的文件体积显著减小,而且能提供数字版权保护,支持流式播放,是 MP3 的有力竞争对手。

参考文献

1. http://en.wikipedia.org/wiki/Comparison_of_audio_formats, 2011-2-26

2. Pohlman Ken C. Principles of digital audio. 4th ed. McGraw-Hill Companies Inc, 2000（张福炎）

quanqiu daohang weixing xitong jieshouqi

全球导航卫星系统接收器(global navigation satellite system receiver, GNSS receiver)

接收全球导航卫星系统(GNSS)发送的信号以确定地面空间位置的一种跟踪定位设备。GNSS 发送的信号是一种可供无数用户共享的信息资源。对于陆地、海洋和空间的广大用户,只要拥有能够接收、跟踪、变换和测量 GNSS 信号的接收设备,即全球导航卫星接收器,就可确定地面空间位置。

1978 年美国发射了第一颗全球定位系统(GPS)卫星。GPS 系统是由美国国防部设计和资助的精巧卫星导航系统,包含了 24 颗能持续发送地理位置海拔高度和时间信号的卫星,这些卫星平均分布运行在六个轨道上。一般来说,在地面上的 GPS

接收器能接收 5～12 个卫星信号，为了获得地面上的定位坐标，至少需要 4 个卫星信号，三个用来确定 GPS 接收器的纬度、经度和海拔高度，第四个则提供同步校正时间。

按接收器的用途可分为：

（1）导航型接收器　主要用于运动载体的导航，实时给出载体的位置和速度。

（2）测地型接收器　主要用于精密大地测量和精密工程测量。

（3）授时型接收器　主要利用 GNSS 卫星提供的高精度时间标准进行授时，常用于天文台及无线电通信中的时间同步。

按接收器工作原理可分为：

（1）码相关型接收机器　利用码相关技术得到伪距观测值。

（2）平方型接收器　利用载波信号的平方技术和测定接收机内产生的载波信号与接收到的载波信号之间的相位差，测定伪距观测值。

（3）混合型接收器　综合上述两种接收机的优点，既可以得到码相位伪距，也可以得到载波相位观测值。

（4）干涉型接收器　将 GNSS 卫星作为射电源，采用干涉测量方法，测定两个被测站间距离。

接收器主要由天线、主机、存储器、高精度高灵敏度的核心程序和电源五部分组成。其中，天线接收来自 20 000 km 左右高空飞行的卫星发射的信号，信号电平、输入功率信噪比均很微弱，加上高空电离层的干扰波，因此，有用信号源淹没在噪声中。为了提高信号强度，一般在天线后端设有前置放大器，以保证接收机对信号进行跟踪、处理和量测。主机包括变频器及中频接收放大器，用于搜索卫星、索引并跟踪卫星、解调出广播电文的信号通道，进行伪距测量、载波相位测量及多普勒频移测量的部件。存储器用于存储卫星星历、卫星历书、接收器采集到的码相位伪距观测值、载波相位观测值及多普勒频移。

随着 GNSS 系统的不断改进，软、硬件的不断完善，应用领域正在不断地开拓，目前已遍及国民经济的各部门，并开始逐步深入人们的日常生活。随着卫星导航定位设备的小型化甚至芯片化，嵌入式 GNSS 产品越来越广泛地应用到人们生活的各个方面。

参考文献

刘基余．GPS 卫星导航定位原理与方法．北京：科学出版社，2005 年

（韩承德）

quanqiu dingwei xitong

全球定位系统（global positioning system，GPS）　利用在轨卫星向位于地球上任意点的接收机发送无线电信号，并通过三边测量方法和应用软件的数值计算，对地球上相关物体进行定位的一种中距离圆形轨道卫星导航系统。又称全球卫星定位系统。GPS 可以为地球表面 98% 的地区提供准确的定位、测速和高精度的定时服务。

美国国防部从 1973 年开始研制 GPS，耗资 300 亿美元，于 1993 年完成全球覆盖率高达 98% 的 24 颗卫星的发射，建成号称第三大航天计划的宏伟工程，它由三大部分组成：

（1）空间部分，由 24 颗卫星组成，其中工作卫星 21 颗，备用卫星 3 颗，它们均匀分布在 6 个相互夹角为 60° 的轨道平面内，即每个轨道上有 4 颗卫星。卫星高度离地面为 20 200 km，绕地球运行一周的时间是 12 恒星时，即一天绕地球两周。卫星的这种空间配置，保证了地球上任何地点任何时间都可以至少同时观测到 4 颗卫星，形成全球性、全天候、三维定速定时的定位系统。

（2）地面控制部分，由一个主控站（位于美国科罗拉多州谢里弗尔空军基地）、一个备用主控站（位于马里兰州盖茨堡）、4 个注入站（分别位于太平洋的卡瓦加兰岛、印度洋的狄哥·伽西亚、大西洋的阿松森岛和美国本土科罗拉多州的科罗拉多斯普林斯）、6 个监测站（除位于上述 4 个地点外，再加上夏威夷和卡纳维拉尔角）以及通信辅助系统组成。地面控制部分的任务是通过应用软件计算出每颗 GPS 卫星在任一时刻的空间位置，实现对 GPS 卫星运行的监控。

（3）用户装置部分，由 GPS 信号接收机和应用软件组成。接收机的任务是跟踪接收 GPS 卫星发射的信号，测量出卫星到接收机天线的传播时间，解释卫星发送的导航信息等，然后通过应用软件实时地计算出所在的三维空间位置。如果接收到 3 颗卫星的信号，应用软件可以利用 3 球定位原理，计算出接收机所在位置的经度、纬度和高程。如果接收到 4 颗或者更多卫星的信号，接收机还可以通过应用软件获得定位时间及更高的定位精度，从而达到精确定位的目的。用于一般导航的 GPS 接收机，体积小、耗电少、操作方便，其定位精度约为 10m 级别；用于精密大地测量的接收机，结构复杂、价格昂贵，其定位精度达厘米（cm）级别。

当今，除了美国的 GPS，还有俄罗斯的 GLO-

NASS 以及正在建设中的欧洲伽利略全球卫星导航系统(GALILEO)和中国的北斗二代(BD)。2007 年中国的北斗导航卫星发射成功,预计在 2020 年形成覆盖全球的卫星导航定位系统。

GPS 的主要用途有:①陆地应用,包括车辆导航、地球观测、地球物理勘探、地壳运动监测、地质灾害预报、大地控制网建立、工程测量、出租车调度与管理等;②海洋应用,包括船舶调度与导航、海面变化监测、海洋石油钻井定位、海洋救援、海洋资源探测等;③航空航天应用,包括飞机导航、航空遥感姿态控制、导弹制导、低轨卫星定轨、航空救援等。

当前,卫星定位技术的主要研究热点是与应用软件紧密相关的网络 RTK(real time kinematic)和精密单点定位技术,尤其是利用网络 RTK 技术,在较大区域内建立连续运行基准站网系统,实现全天候、全自动和实时地为用户提供不同精度的定位导航信息。对于精密单点定位技术,主要研究非差相位精密单点定位及其实现方法。由多种卫星导航系统构成的组合导航技术和网络化导航系统,由于不受单一系统的制约,能有效提高卫星导航定位的可靠性和精度,具有重要的发展和应用前景。

参考文献

周忠谟,易杰军. GPS 全球定位系统原理及其应用. 北京:测绘出版社,1997

(黄杏元)

quanwen jiansuo

全文检索(full-text retrieval) 把文献中出现的每一个词(或字)都作为检索入口的基于全文标引的检索过程和技术。在全文检索系统中,文献中任何有检索意义的词或字符串都可被检索出来。可以用布尔逻辑进行组合检索;可以用位置逻辑进行组合检索;可以使用停用词技术剔除文献中无检索意义的词,留下有实际意义的词作为检索入口而加以标引。若只能根据文献的外部特征找出全文,或只提供布尔逻辑运算,而没有提供位置逻辑运算,都不能称作全文检索。

1959 年美国法律界在匹兹堡大学建立了第一个全文检索系统。广域信息服务系统是新一代全文检索系统的典型。

全文检索与传统的人工标引检索相比较,具有以下优点:①自动建立索引库,速度快;②不存在词汇滞后问题;③不损失专指性;④不产生漏检;⑤可以直接提供原文献。

与针对二次文献的检索相比,全文检索的突出优点是可以检索原文中的全部内容,但由于计算量大幅度提高,也会导致索引占用的物理空间大,检索速度慢。但随着计算机软、硬件的发展及数据压缩技术的成熟,全文检索技术已经完全实用,互联网上的各种资源数据库普遍采用先进的全文检索技术。

目前国际上比较流行的全文检索模型主要有:倒排表模型、署名文件、位图、PAT 树和 PAT 数组。前三种模型属于同一基本观念的变体,即把文档看成索引项的集合,所以,这就要求索引数据必须具有文档-索引项的结构,也只能实现简单的查询。PAT 树和 PAT 数组将索引数据看成一组半无限串的叠加,从而避免了这个弱点。

国内对全文检索模型的研究也很多,相邻矩阵模型和互关联后继树模型都是针对中文提出的全文检索模型。这两种模型都能够利用索引生成原文,因而可以完全抛弃原文,节约存储空间。

为了提高全文检索的结果质量,采用相关排序与相关反馈等技术。全文检索的扩展包括能利用文字来检索多媒体信息,结合**超文本**技术及通过交互式浏览和导航改善检索效果。

目前世界上还没有统一的标准对全文检索模型进行比较全面的功能与性能评估。比较常用的标准有:

(1)时间复杂度 包括创建索引的时间效率以及检索的时间效率。对于索引模型更重要的是检索性能,所以,更常考虑的是模型的检索速度。

(2)空间复杂度 由于现在索引数据的海量性,索引模型中任何能够减少索引空间消耗的机制均可产生显著的实用效果。

(3)查询完备性 一个好的全文检索模型应该能够实现多种查询,如字符串查询,前缀查询,范围查询。

(4)适应性 好的索引模型应该可以处理各种不同类型具有不同特点的数据,如长度相差很大的记录,或者没有文档关键词结构的数据。

(5)动态性 索引模型最好能适应数据的动态改变,包括关键词词典中词条的增加,或索引记录的改变等。

中文全文检索可分为按字全文检索与按词全文检索。按字全文检索具有检索速度快、空间开销小等优点。同时,如果要利用较为高级的检索技术,如相关排序,则按词建立索引库和检索具有较大的优越性。

参考文献

1. Salton G. Automatic text processing: the trans-

formation, analysis, and retrieval of information by computer. Addison Wesley, 1989

2. Frakes W B, Yates R B. Information retrieval: data structures and algorithms. Prentice Hall,1992

3. 刘挺,秦兵,张宇,车万翔. 信息检索系统导论. 北京:机械工业出版社,2008

(刘挺　邵艳秋　施水才)

quanxi cunchuqi

全息存储器(holographic storage)　以激光器为光源,利用全息照相的方法,实现数字信息存储的装置。全息存储的基本原理可简单描述为待存储的二值数据以页的方式映射到振幅型空间光调制器(SLM)上形成一个二维信息页加载到信号光上,然后与参考光在记录介质中发生干涉,利用材料的光折变效应形成体全息光栅,从而完成信息的记录。读出时使用与存储时所用参考光相同的光束寻址,可以读出相应的存储在晶体中的全息干涉图,然后使用光耦合器件(CCD)对参考光读出的图像进行信号处理和采集。当全息图被存储在厚的材料中时,数据的读出对参考光的特性十分敏感,这如同一个可调的结构,存储材料的厚度越厚,记录结构越可调,通过改变参考光的特性,例如参考光的入射角或波长,根据体全息图的布拉格(Bragg)角度的选择性或波长的选择性,就可以存储另一幅全息图,从而在同一存储体内存储许多不同的数据页,这种技术称为复用。采用角度复用的系统中,多幅全息图以不同的参考光入射角或不同的波长记录在同一存储体中。所存全息图的数目与记录材料的厚度及特性有关。体全息存储系统的体积密度理论上可以达到$1/\lambda^3$,其中λ为光波波长。改变参考光的入射角度或波长可以实现多重存储。通过使用复用技术可以增加存储密度,显著提高整体存储容量,这正是体全息存储区别于传统存储技术之处。目前SLM和CCD器件的飞速发展,为充分发挥体全息存储的优点奠定了基础,使之成为下一代最有可能实用化的新型存储技术的代表。

自从1948年英国人Dennis Gabor为提高电子显微镜的分辨率发明了全息术以来,在其后的60多年的时间内,美国、日本、英国和中国先后展开了对体全息存储的研究。特别是进入20世纪80年代,光学计算研究的热潮重新激起人们对全息存储的兴趣。另外存储工业以外的电子消费、医学、娱乐工业推动了全息存储系统所需的低廉、可靠且实用部件的极大发展,促进了体全息存储的研究。1995年在美国国家存储工业联合会(NSIC)组织下由12个科研、大学单位参加的联合研究体协作组织,投资约7000万美元,实施"光折变信息存储材料"(PRISM)和"全息数据存储系统"(HDSS)这两个研究项目,从存储材料、关键器件、光学系统结构、信号处理等几方面对体全息存储技术展开有序的全方位研究。2011年,通用电气(GE)宣称已成功开发新型微全息存储材料和技术。新开发的微全息存储材料碟片存储容量500 GB,存储速度与蓝光光碟相当。GE科研人员将继续开发存储容量超过1000 GB的微全息存储碟片,并积极推动商用化。

体全息存储因具有高存储容量、数据并行传输、冗余度高、寻址速度快、具有关联寻址等特性,不但可用于数字全息存储器,还可以用于一些光学处理系统和图像识别系统。其应用领域将包括卫星通信、空中侦察、高速数字图书馆、战术车辆信息的可靠存储以及娱乐、医学和军事用途的图像处理等,范围十分广泛。体全息存储跨越了信息存储和信息处理两个领域,随着各种相关技术(如光源、材料和关键光器件)的飞速发展,体全息存储必将在计算机科学和光学信息处理中发挥重要作用。

参考文献

1. Yeh P, Gu C. Landmark papers on photorefractive nonlinear optics. Singapore: World Scientific, 1995

2. Coufal H J, Psaltis D, Sincerbox G T. Holographic data storage. New York: Springer-Verlag, 2000

(何庆声　全国藩　吴非)

quesheng luoji

缺省逻辑(default logic)　以缺省命题作为理论完备化手段的一种**非单调逻辑**。R. Reiter提出的缺省逻辑是非单调逻辑的一种。非单调逻辑可以体现现实中常见的现象:当新的事实被认识时,原先的某些结论可能要被推翻。缺省逻辑的基本思想为:若没有事实证明S不成立,则认为S在缺省条件下成立。由于人们对客观事物的认识是渐进的,所以推理时所依赖的**知识库**也是逐步扩充的。为了使推理在当前不丰富的知识库(称作不完备的理论)下也能够进行,可以"想当然"地对知识库进行扩充(称作理论的完备化)。扩充出来的内容是缺省知识,它只能保证不与知识库的其他部分矛盾,根据它们推出的是对现实世界的一种猜测。

缺省逻辑提供一组缺省命题，作为理论的完备化的手段。缺省命题的形式为：

前提 α：缺省要求 $\beta_1,\cdots,\beta_n \to$ 结论 γ

其中，α,β_i,γ 均为一阶谓词逻辑的合适公式。上式读为：若不能证明 β_i 中的任何一个不成立，则从 α 可以推出 γ。通常只考虑闭的缺省命题(即 α,β_i,γ 均为闭的合适公式)。

令 D 为一组缺省命题，W 为一组闭的合适公式，二元组 $\Delta=(D,W)$ 称为一个缺省理论。考虑如下的扩张算子 Γ：对于任意合适公式集 S，$\Gamma(S)$ 是具有如下三条性质的最小集合：

(1) 包含 W；

(2) 在普通命题逻辑的推理下封闭；

(3) 对 D 中任何缺省命题，若 $\alpha\in\Gamma(S)$，$\sim\beta_1,\cdots,\sim\beta_n\notin S$，

则 $\gamma\in\Gamma(S)$。

扩张算子 Γ 的一个不动点 E(即 $\Gamma(E)=E$)称为缺省理论 Δ 的一个扩张，它是用 D 对 W 推理后所得到的一个可能的世界全景。

缺省逻辑具有非单调的特征：若在 $\Delta=(D,W)$ 中将 W 扩充为 W' 而产生 $\Delta'=(D,W')$，(即新的事实被认识)，则在 Δ' 的扩张中可能会排除原先 Δ 的扩张中的某些公式(原先的某些结论被推翻)。扩充缺省命题集 D 也会产生类似的后果。除了非单调性，一般的缺省理论还可能出现扩张不存在，扩张不唯一等现象。这些都是实用中的巨大障碍。为此需要考察特殊的具有实用意义的缺省理论。Reiter 本人提出的正规缺省理论就是其中的一种。

在正规缺省理论中，缺省命题只能取如下的形式：

前提 α：缺省要求 $\beta\to$ 结论 β

对于正规缺省理论，我们有以下更加实用的结论：

(1) 每个正规缺省理论都有一个扩张。

(2) 扩充正规缺省理论的缺省命题集 D 不会丢失任何原先已经推出的命题。

(3) 正规缺省理论若有两个扩张 E 和 E'，则 $E\cup E'$ 一定是不一致的。或者说，如果正规缺省理论 $\Delta=(D,W)$ 中 D 的推论集和 W 的并是一致的，则 Δ 只有唯一的扩张。

参考文献

陆汝钤. 人工智能(下). 北京：科学出版社，2002

（刘椿年）

qun

群(group) 一种具有一个二元运算的代数结构。

若非空集合 G 及其上的二元运算 $*$ 满足结合律，即对任意 $a,b,c\in G$ 有 $a*(b*c)=(a*b)*c$，则称 G 为一个**半群**，$*$ 为半群 G 的乘法，$a*b(a,b\in G)$ 为 a 与 b 的积，常简写为 ab。

由归纳法可证明，对任意 $a_1,\cdots,a_n\in G(n\geqslant 1)$ 有唯一确定的积 $a_1\cdots a_n$，即只要 $a_1,\cdots,a_n$ 的次序不变，无论在它们间怎样添加括号，结果都一样。这就是所谓的广义结合律。

若 $e\in G$ 使得对任意 $a\in G$ 有 $ae=a=ea$，则称 e 为半群 G 的一个幺元或单位元。具有幺元的半群称为幺半群。

全体正整数关于普通的加法和乘法分别构成半群和幺半群。全体实 $n(n\geqslant 1)$ 阶方阵关于矩阵乘法构成幺半群。

如果半群 G 满足以下公理：

(1) 存在左幺元，即对每个 $a\in G$ 皆有 $ea=a$；

(2) 存在左逆元，即对每个 $a\in G$，皆有 $a^{-1}\in G$ 使 $a^{-1}a=e$；

则称 G 为一个群，记为 $\langle G,*\rangle$。

可以证明，对群 G 中任意元素 a 有 $ae=a$ 和 $aa^{-1}=e$，而且 e 和 a^{-1} 还都是唯一的，因此称 e 为 G 的幺元或单位元，a^{-1} 为 a 的逆元。此外，对任意 $a,b\in G$ 显然有 $(ab)^{-1}=b^{-1}a^{-1}$。

根据广义结合律，对任意 $a\in G$ 和正整数 n，可令

$$a^0=e,\quad a^n=\underbrace{aa\cdots a}_{n\text{个}},\quad a^{-n}=(a^{-1})^n$$

于是，对任意整数 m,n 有 $(a^m)^n=a^{mn}$ 和 $a^ma^n=a^{m+n}$。

如果 G 为有穷集合，则称群 G 为有限群，否则称群 G 为无限群。G 中元素个数称为群 G 的阶，记为 $|G|$。

若有 $a\in G$ 使 $G=\{a^n\mid n=0,\pm1,\pm2,\cdots\}$，则称 G 为**循环群**，a 为 G 的一个**生成元**，记为 $G=\langle a\rangle$。

设 H 为 G 的非空子集。若对任意 $a,b\in H$ 有 $ab^{-1}\in H$，则称 H 为 G 的一个**子群**，记为 $H\leqslant G$。$\{e\}$ 为 G 的子群，称为单位子群。若 $H\leqslant G$ 且 $H\neq G$，则称 H 为 G 的真子群。

设 $H\leqslant G$。对任意 $a\in G$，称 $aH=\{ah\mid h\in H\}$ 和 $Ha=\{ha\mid h\in H\}$ 为 G 关于 H 的左陪集和右陪集。G 关于 H 的不同左(右)陪集都互不相交且 $\overline{\overline{\{aH\mid a\in G\}}}=\overline{\overline{\{Ha\mid a\in G\}}}$，并称 G 关于 H 的不同左陪集个数为 H 对 G 的指数，记为 $[G:H]$。

设 $H\leqslant G$，若对任意 $a\in G$ 有 $aH=Ha$，则称 H 为

G 的一个**正规子群**，记为$H \trianglelefteq G$。这时，对任意 $a,b \in G$ 必有

$$aH \circ bH \triangleq \{ah * bh' | h,h' \in H\} = abH$$

因此$\{aH | a \in G\}$关于上面定义的左陪集乘法 ∘ 成群，称为 G 关于 H 的**商群**，记为 G/H。

如果群 G（幺半群或半群）的乘法 * 满足交换律，即对任意 $a,b \in G$ 有 $ab = ba$，则称 G 为交换的。交换群又称**阿贝尔群**。这时，常将 * 改作 +，并相应地把“乘法”、“积”、“幺元 e”、“逆元 a^{-1}”和“商 ab^{-1}”改称“加法”、“和”、“零元 0”、“负元 $-a$”和“差 $a-b$”。

循环群都是交换群。

若 A 为非空集合，则$\{f | f: A \to A$ 为双射$\}$关于函数合成构成群，称为 A 上的**变换群**。$A = \{1,2,\cdots,n\}$（$n \geqslant 1$）上的变换群称为 n 次**对称群**，用 S_n 表示。S_n 的子群称为 n 次**置换群**。变换群、对称群和置换群一般不是交换群。

设 G 和 G^* 为两个群。若 $f: G \to G^*$ 对任意 $a,b \in G$ 有 $f(ab) = f(a)f(b)$，则称 f 为一个**群同态**，并称 $\ker(f) = \{a \in G | f(a)$ 为 G^* 的幺元$\}$为 f 的同态核。若群同态 f 为双射，则称 f 为**群同构**，并称 G 与 G^* 同构，记为 $G \cong G^*$。

每个群都与某个变换群同构，每个有限群都与某个置换群同构。

设 G 和 G^* 为两个群。若 $f: G \to G^*$ 为群同态，则 $\ker(f)$ 为 G 的正规子群，而且当 $f(G) = G^*$ 时有 $G/\ker(f) \cong G^*$（**群同态定理**）。

半群、幺半群和群是数学及其应用中最基本的概念之一，已渗透到现代数学的许多分支并起着重要作用，还形成了一些新学科。此外，它们在理论物理、结晶学，特别是计算机科学（诸如形式语言、自动机理论和编码理论等）中，都有重要应用。

参考文献

1. Hangerford T W. 代数学. 冯克勤，译. 长沙：湖南教育出版社，1985

2. 张远达. 有限群构造（上、下）. 北京：科学出版社，1984

（王兵山　王水汀）

qunti donghua

群体动画（crowd animation）　用计算机模拟在同一物理环境下拥有相同目的的一群个体的行为的技术，群体的行为有别于作为单独个体时的行为。

影视作品中千军万马的宏伟场面常常能给观众留下极为深刻的印象。当人们看到《阿凡达》、《指环王Ⅱ》、《特洛伊》中那些浩浩荡荡的军队、惊心动魄的作战场景时，很少能不为其强烈的视觉效果所震撼。用传统方法摄制这类宏大场面，需招聘大量的群众演员按照剧情进行演练。这不仅要耗费大量的人力、物力和财力，而且场地选择、指挥调度亦并非易事。群组动画的出现，为解决这些问题提供了一个不受时空限制的方法。它不仅可节省人力成本，还能极大地增强影视作品的艺术表现力。1987年，Reynolds 首先提出了一个分布式的行为模型来模拟群组的行为，例如真实世界中的鸟群、鱼群或其他动物群体，奠定了群组动画的研究基础。这个群组基于粒子系统的形式进行模拟，每个粒子表示一个个体，这个模型又被称为 Boids 模型。每个独立的群组粒子对周围的动态环境进行独立感知，然后按局部控制规则进行运动，使得粒子满足三条原则：与其他粒子和环境的碰撞避免，与邻近粒子的速度匹配，向群组质心的移动趋势。该先驱性的工作由分离、聚合和排列三大原则来生成复杂的群组动画，成为群组模拟的常用模型之一。1999 年，Reynolds 提出了驾驭行为模型。Reynolds 指出，一个自主角色的行为包括三个抽象层次，由高到低分别是动作选择、控制行为和关节运动。Reynolds 总结了作为中间层次控制行为的若干种类，如排斥、追逐等。他同时提出，群体的整体特征必须通过个体的行为来表现，因此对整体的行为控制应该转换为对个体的作用，最终成为影响个体运动的一个因素。更一般地，群组动画的研究内容包括群组模型、基于力场的行人动态建模、数据驱动的群组模拟、大规模群组场景的快速绘制、人群疏散模拟、个体运动的合成、个体之间的碰撞检测和碰撞避免、群组导航和运动控制、群组中的感知计算、**人工生命**、基于 GPU 的人群模拟、基于视觉的人群生成、群体场景创作、汽车群组模拟、群组特效模拟等。

群组模拟软件包括 Weta Digital 公司的 Massive、Softimage | behavior 等。Massive 是为《指环王》三部曲的制作而开发的，其创始人为新西兰人 Stephen Regelous。Regelous 担任《指环王》三部曲的群组模拟主管，他花了 2 年的时间编写出了 Massive，并因这方面的杰出贡献荣获 2003 年奥斯卡科学与工程奖。

参考文献

1. 彭群生，金小刚，万华根，等. 计算机图形学应用基础. 北京：科学出版社，2009

2. http://en.wikipedia.org/wiki/Crowd_simulation

（金小刚）

R

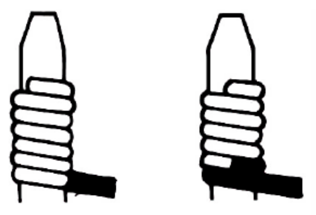
raojie

绕接(wire-wrap connection) 用绕接工具把单股导线剥去外皮露出导体,并将导体紧绕在接线端子上,以达到电气连接的工艺过程。在导线环绕拉力作用下,导线与端子的锐利棱边接触,在接触处产生约 1.03×10^9 Pa (10 500 kgf/cm^2)的强大压力,使导线和金属端子上的氧化层被破坏,形成清洁的金属接触,并有良好的气密性,能避免再受到氧化和腐蚀,并使连接获得优良的导电性能。试验表明,绕接点可以保证 40 年以上的可靠连接。

绕接技术是由贝尔电话实验室发明的,1952 年首先用在纵横制自动电话交换机上,不但提高了机器的可靠性,节省了各种工艺材料,还缩短了连线的施工时间。

绕接只适用于单股导线,或采用镀锡工艺将多股聚合成一股的导线,绕接线的直径一般从 0.25 ~ 1.3 mm。适合绕接的端子必须具有两个以上的锐角,一般有六种形状:方形、U 形、V 形、滚花扁平形、滚花菱形、双三角形等。在计算机底板接线中常用的是方形。

绕接有两种形式。图 1(a)所示为标准绕接,即常规型绕接。它是将剥去绝缘层的导线,以紧密螺旋方式牢固地缠绕在端子上,一般适用于直径 0.5 mm 以上的导线。图 1(b)为防震型绕接,即除金属部分外,还将一部分未剥绝缘层的导线也缠绕在端子上,绝缘层部分至少绕有三个以上的棱角,以达到良好的防震性。



(a) 标准绕接　　(b) 防震绕接

图 1　绕接点的种类

绕接在 20 世纪 70 年代已推广应用于计算机、通信设备、家用电器以及各种电气控制装置中,绕接用的工艺设备、工具、线材等已不断完善配套,并制订了绕接线材、端子、绕接质量检验等各种标准。绕接可用微型计算机编程全自动进行,定位速度达 10 in/s(1 in = 25.4 mm)。 (顾本斗)

re sheji

热设计(thermal management, thermal design) 对计算机系统中的电子元件、电路模块、设备等的工作温度和温度梯度加以经济、合理、有效的控制,并考虑温度和噪声对人的影响而进行的设计。

电子元器件的性能与温度有关,其故障率随温度的升高成指数关系增加。近年来,大规模和超大规模集成电路的迅速发展,使计算机的热通量(单位为 W/cm^2)和热密度(单位为 W/cm^3)越来越高;同时为了提高可靠性,元器件的工作温度又应有所降低;这些都对热设计提出更高的要求。热设计的目的是将众多元器件产生的热量有效地转移出去,以提高系统的可靠性。在热量转移的路径中存在热阻。计算机中的热阻一般可分为 3 级:

(1) 元件级热阻(即内热阻)　元器件节点与元器件表面(或散热器表面)上参考点间的热阻。系统热设计人员通常无法改变元器件的内热阻。

(2) 插件级热阻元器件　表面参考点与对流层中某指定点间的热阻。

(3) 系统级热阻　插件与最终换热器之间的热阻。最终换热器可能是计算机房的空调系统或冷水机组。

为了达到计算机系统正常工作所要求的温度和温度梯度,需要进行系统的热设计。热设计同时要考虑到可靠性指标(参见**可靠性设计**),安装空间限制,维护以及成本等因素。这些因素往往是相互矛盾的,需要经过一系列技术方案的分析比较折中之后才能确定最佳选择。热设计的重点在于选择冷却形式并进行设计和计算。

计算机的冷却形式有空气自然对流冷却、空气强迫对流冷却、水强迫对流冷却和相变冷却(如碳氟化物自然对流、碳氟化物沸腾)等。图 1 表示各种冷却形式的适用范围。

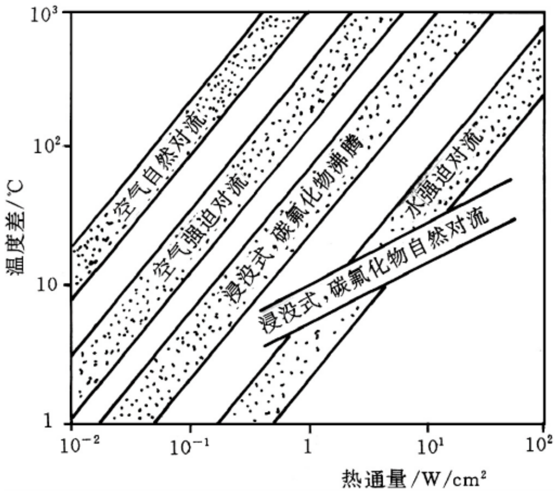


图 1　各种冷却形式的适用范围

（1）空气强迫对流冷却　用风机加压使空气流过机柜带走热量。此形式结构简单,维护方便,可靠性高,成本低,目前国内外采用较多。空气强迫对流冷却可分为长、短风路两种。在长风路形式中空气进入机柜后经过几层插件才排出机柜,如图 2 所示。短风路形式则是空气进入机柜后只经过一层插件即排出机柜,如图 3 所示。短风路形式机柜内温度梯度小;长风路形式风量大,可提高组装密度。

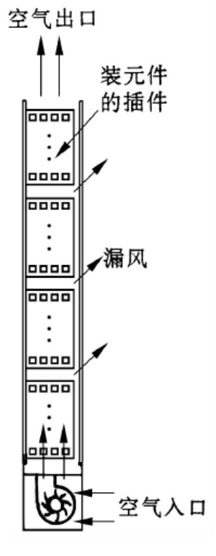


图 2　长风路形式

（2）水强迫对流冷却　用水代替空气作冷却介质,换热系数可提高 1～2 数量级。因水有腐蚀性且绝缘性能差,通常采用间接的水冷方式,即被冷却的元件不与水直接接触,热量经其他介质传导到水,再经对流带走。水强迫对流冷却能有效地降低元器件的外热阻,且具有噪声小,机柜内温度均匀等优点。

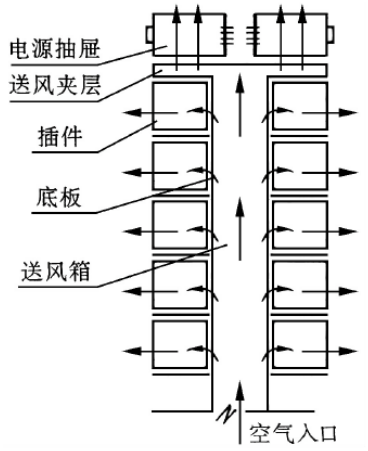


图 3　短风路形式

但它技术较复杂,成本较高,目前多用于大型和巨型机的功率密度高的主机机柜。

（3）间接相变冷却　这种形式采取传导与相变换热相结合的方式。如 Cyber 203/205 机的陶瓷基片是用弹簧卡压紧在冷却管上(参见图 4),冷却管内通氟利昂,氟利昂蒸发将热量带走。这种形式热阻较低,组件的热通量可达 2 W/cm^2。

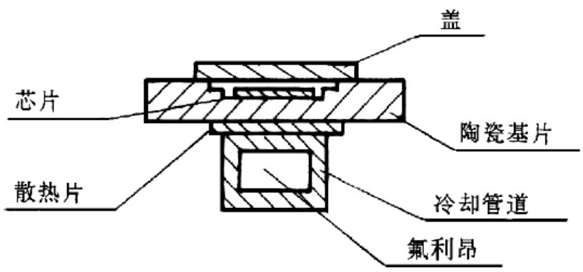


图 4　间接相变冷却形式

（4）直接相变冷却　当功率密度进一步提高时,间接相变冷却不能满足要求,就要采取将元器件直接浸没在冷却液(如碳氟化合物)中的这一非常有效的换热方式。直接相变冷却对冷却液除了要求换热效率高之外,还要求介电强度高,化学惰性好。在结构上要求密封严格,因此成本也高。

（5）热电致冷　这是正在研究但已取得实用进展的冷却技术。根据珀耳帖效应,当直流电通过两种不同材料的节点时,节点将吸热或放热。热电致冷器的电偶是利用特制的 N 型和 P 型半导体材料,用铜连接片焊接而成。单个的电偶冷却能力小,为了增大冷却能力,通常热电致冷器由几十对电偶组成。将热端贴在散热器上,冷端贴在被冷却的元器件上,能使元器件的温度达到或低于环境温度,温差可达 65 ℃,可带走 125 W 的热流量。为了增大温

差,可采用多级热电致冷器,4 级的热电致冷器温差可达 125 ℃。热电致冷器的优点是:改变电流的方向可进行加热或冷却,改变电流的大小可调整换热量,无运动部件,无噪声,在失重或高重力条件下及任何方位均可很好地工作。缺点是体积较大,目前效率仍较低,即输入功率与制冷量之比较高,这种情况需靠发展新的半导体材料得以改善。

(6) 热管　热管是利用毛细管原理,采用介质相变来吸热或放热的真空密封传热器件,其传热能力比同样截面的铜要大几十倍,有资料报道可以排出 47 W/cm^{2}的热通量,在计算机中可用来冷却大功率集成电路。热管的优点是传热能力大,无噪声,但目前价格较高,对真空密封要求很严格(否则很易失效)。 (唐玛琍)

rengong shenjing wangluo

人工神经网络(artificial neural network) 由大量处理单元互连组成的非线性、自适应信息处理系统。它是在现代神经科学研究成果的基础上提出的,试图通过模拟大脑神经网络处理、记忆信息的方式进行信息处理。

人工神经网络具有四个基本特性:

(1)非线性　非线性关系是自然界的普遍特性。大脑的智慧就是一种非线性现象。人工神经元处于激活或抑制两种不同的状态,这种行为在数学上表现为一种非线性关系。具有阈值的神经元构成的网络具有更好的性能,可以提高容错性和存储容量。

(2) 非局域性　一个神经网络通常由多个神经元广泛连接而成。一个系统的整体行为不仅取决于单个神经元的特征,而且可能主要由单元之间的相互作用、相互连接所决定。通过单元之间的大量连接模拟大脑的非局域性。联想记忆是非局域性的典型例子。

(3) 非定常性　人工神经网络具有自适应、自组织、自学习能力。神经网络不但处理的信息可以有各种变化,而且在处理信息的同时,非线性动力系统本身也在不断地变化。经常采用迭代过程描写动力系统的演化过程。

(4) 非凸性　一个系统的演化方向,在一定条件下将取决于某个特定的状态函数,例如能量函数,它的极值相应于系统比较稳定的状态。非凸性是指这种函数有多个极值,故系统具有多个较稳定的平衡态,这将导致系统演化的多样性。

人工神经网络中,神经元处理单元可表示不同的对象,例如特征、字母、概念,或是一些有意义的抽象模式。网络中处理单元的类型分为三类:输入单元、输出单元和隐单元。输入单元接受外部世界的信号和数据;输出单元实现系统处理结果的输出;隐单元是处在输入与输出单元之间,不能由系统外部观察的单元。神经元间的连接权值反映了单元间的连接强度,信息的表示和处理体现在网络处理单元的连接关系之中。人工神经网络是一种非程序化、适应性、大脑风格的信息处理,其本质是通过网络的变换和动力学行为得到一种并行分布式的信息处理功能,并在不同程度和层次上模仿人脑神经系统的信息处理功能。它是涉及神经科学、思维科学、**人工智能**、计算机科学等多个领域的交叉学科。

人工神经网络是并行分布式系统,采用了与传统人工智能和信息处理技术完全不同的机理,克服了传统的基于逻辑符号的人工智能在处理直觉、非结构化信息方面的缺陷,具有自适应、自组织和实时学习的特点。

历史沿革　1943 年,心理学家 W. S. McCulloch 和数理逻辑学家 W. Pitts 建立了神经网络的数学模型,称为 MP 模型。他们通过 MP 模型提出了神经元的形式化数学描述和网络结构方法,证明了单个神经元能执行逻辑功能,从而开创了人工神经网络研究的时代。1949 年,心理学家提出了突触联系强度可变的设想。20 世纪 60 年代,人工神经网络得到了进一步的发展,更完善的神经网络模型被提出,其中包括**感知机**和自适应线性元件等。M. Minsky 等仔细分析了以感知机为代表的神经网络系统的功能及局限后,于 1969 年出版了"Perceptron"一书,指出感知机不能解决高阶谓词问题。他们的论点极大地影响了神经网络的研究,加之当时串行计算机和人工智能所取得的成就,掩盖了发展新型计算机和人工智能新途径的必要性和迫切性,使人工神经网络的研究处于低潮。在此期间,一些人工神经网络的研究者仍然致力于这一研究,提出了**自适应谐振理论**(ART 网)、**自组织映射**、认知机网络,同时进行了神经网络数学理论的研究。以上研究为神经网络的研究和发展奠定了基础。1982 年,美国加州工学院生物物理学家 J. J. Hopfield 提出了 Hopfield 神经网络模型,引入了"计算能量"概念,给出了网络稳定性判断。1984 年,他又提出了连续时间 Hopfield 神经网络模型,为神经计算机的研究做了开拓性的工作,开创了神经网络用于联想记忆和优化计算的

新途径，有力地推动了神经网络的研究（参见**Hopfield 网络**）。1985 年，又有学者提出了**Boltzmann 机**模型，在学习中采用统计热力学模拟退火技术，保证整个系统趋于全局稳定点。1986 年进行认知微观结构的研究，提出了并行分布处理的理论。人工神经网络的研究受到了各个发达国家的重视，美国国会通过决议将 1990 年 1 月 5 日开始的 10 年定为“脑的十年”，国际研究组织号召它的成员国将“脑的十年”变为全球行动。在日本的“真实世界计算”（RWC）项目中，人工神经网络的研究成了一个重要的组成部分。

基本内容 人工神经网络模型主要考虑网络连接的拓扑结构、神经元的特征、学习规则等。目前，已有近 40 种神经网络模型，其中有**反向传播网络**、**感知机**、自组织映射、霍普菲尔特网络、玻耳兹曼机、自适应谐振理论等。根据连接的拓扑结构，神经网络模型可以分为：

（1）前向网络 网络中各神经元接受前一级的输入，并输出到下一级，网络中没有反馈，可以用一个有向无环路图表示。这种网络实现信号从输入空间到输出空间的变换，它的信息处理能力来自于简单非线性函数的多次复合。网络结构简单，易于实现。反传网络是一种典型的前向网络。

（2）反馈网络 网络内神经元间有反馈，可以用一个无向的完备图表示。这种神经网络的信息处理是状态的变换，可以用动力学系统理论处理。系统的稳定性与联想记忆功能有密切关系。霍普菲尔特网络、玻耳兹曼机均属于这种类型。

学习是人工神经网络研究的一个重要内容。它的适应性是通过学习实现的。根据环境的变化，对权值进行调整，改善系统的行为。由 Hebb 提出的 Hebb 学习规则为神经网络的学习算法奠定了基础。Hebb 规则认为学习过程最终发生在神经元之间的突触部位，突触的联系强度随着突触前后神经元的活动而变化。在此基础上，人们提出了各种学习规则和算法，以适应不同网络模型的需要。有效的学习算法，使得神经网络能够通过连接权值的调整，构造客观世界的内在表示，形成具有特色的信息处理方法，信息存储和处理体现在网络的连接中。

根据学习环境不同，神经网络的学习方式可分为**监督学习**和**非监督学习**。在监督学习中，将训练样本的数据加到网络输入端，同时将相应的期望输出与网络输出相比较，得到误差信号，以此控制权值连接强度的调整，经多次训练后收敛到一个确定的权值。当样本情况发生变化时，经学习可以修改权值以适应新的环境。使用监督学习的神经网络模型有**反向传播网络**、**感知机**等。非监督学习时，事先不给定标准样本，直接将网络置于环境之中，学习阶段与工作阶段成为一体。此时，学习规律的变化服从连接权值的演变方程。非监督学习最简单的例子是 Hebb 学习规则。竞争学习规则是一个更复杂的非监督学习的例子，它是根据已建立的聚类进行权值调整。自组织映射、自适应谐振理论网络等都是与竞争学习有关的典型模型。

研究神经网络的非线性动力学性质，主要采用动力学系统理论、非线性规划理论和统计理论，来分析神经网络的演化过程和吸引子的性质，探索神经网络的协同行为和集体计算功能，了解神经信息处理的机制。为了探讨神经网络在整体性和模糊性方面处理信息的可能，混沌理论的概念和方法将会发挥作用。

发展趋势 人工神经网络特有的非线性适应性信息处理能力，克服了传统**人工智能**方法对于直觉，如模式、语音识别、非结构化信息处理方面的缺陷，使之在神经专家系统、**模式识别**、智能控制、组合优化、预测等领域得到成功应用。

人工神经网络与其他传统方法相结合，将推动人工智能和信息处理技术不断发展。近年来，人工神经网络正向模拟人类认知的道路上更加深入发展，与模糊系统、遗传算法、进化机制等结合，形成**计算智能**，成为人工智能的一个重要方向，将在实际应用中得到发展。将信息几何应用于人工神经网络的研究，为人工神经网络的理论研究开辟了新的途径。神经计算机的研究取得进展，已有产品进入市场。光电结合的神经计算机为人工神经网络的发展提供了良好条件。

参考文献

1. Rumelhart D E, McClelland J L. Parallel distributed processing: Explorations in the microstructure of cognition. MIT Press, 1986

2. 史忠植. 神经网络. 北京：高等教育出版社，2009

（史忠植）

rengong shenjing wangluo zai moshi shibie zhong de yingyong

人工神经网络在模式识别中的应用（application of artificial neural networks to pattern recognition） 模式识别（参见**模式识别**）是

人工神经网络应用的一个重要方面，使用的网络形式有很多种。最简单的单层网络就是早期的**感知机**(参见**感知机**)，当样本是线性可分时，用感知机学习算法在有限步内可以收敛到解；也可以用使均方误差(MSE)最小的方法训练感知机，这样即使样本不是线性可分时，它也可收敛于使 MSE 最小的解，此时它可等价于模式识别中常用的 Fisher 线性判别函数。

含有隐层的**多层前向网络**(图 1，其中 W 为权重)可以构造出复杂的分界面，它是模式识别中最常用的网络形式。多层网络结构与分类区的关系见表 1 。

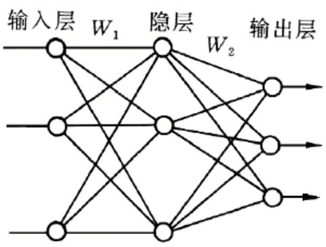


图 1　多层前向网络

由表 1 可见，无隐层的网络只能区分可由超平面分开的类区；单隐层的网络可以形成凸区域；含两个隐层的网络则可形成任意复杂的类区(包括非连通的类区)。

在理想情况下(训练样本和网络规模都不受限制)，多层前向网络的输出可任意逼近样本的后验概率分布，所以它可以趋近最佳分类器(Bayes 分类器)。当训练样本有限时，应当合理选择网络规模(主要是隐层数及各隐层单元数)，以使网络有较好的学习效果和推广能力。

多层前向网络的隐单元的作用函数有许多形式，常用的是 S 形函数(或称 Sigmoid 函数)，其形式为

$$f(x) = \frac{1}{1 + e^{-x}}$$

也可以采用径向基函数(RBF)，它是一种中心对称的函数，例如高斯函数：

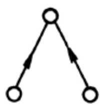
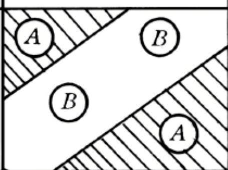
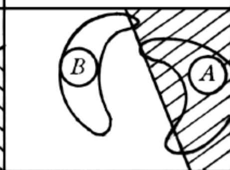
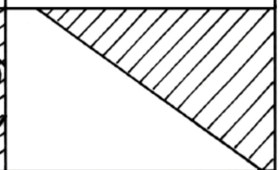
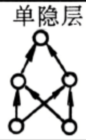
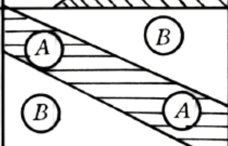
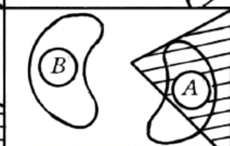
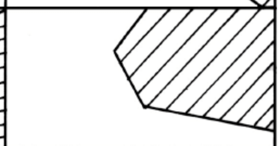
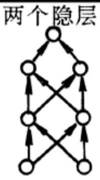
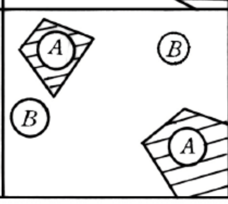
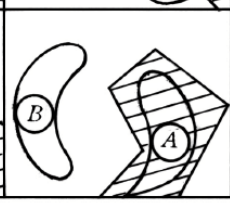
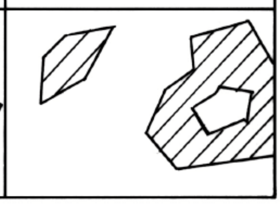
$$f(x) = \frac{1}{\sqrt{2\pi\sigma}}\exp\left(\frac{x - c}{2\sigma}\right)^2$$

它有两个参数，中心位置 c 和宽度 σ，如图 2 所示，此时网络的作用相当于用 Parzen 窗法恢复概率密度。因而是一种非参数法的分类器。

另一种用于模式识别的神经网络是**概率神经网络**，它是一个有两层隐单元的多层前向网络，如图 3，其中第一隐层(模式单元)的作用函数是高斯函数，第二隐层用求和函数，输出单元是求和并判别。适当选择各连接权的值，可使 $f_A(\boldsymbol{X})$，$f_B(\boldsymbol{X})$ 分别代表两类(A 类和 B 类)的用方差为 σ 的正态核函数估计的分布密度，再经加权组合进行判别分类。控制高斯函数方差 σ 的大小，可获得不同类型的分类器，σ 为某固定常数时，相当于恢复分布的分类器，σ 趋近无限大时，相当于分段线性分类器，σ 为零时相当于近邻分类器。

用多层前向网络也可以实现树分类器，图 4 表示一个二值决策树对应的多层前向网络，第一隐层(区分层)的作用是将特征空间划分成不同的开区域，第二隐层(与层)用于形成凸区域，输出层是或层，用于联合各凸区以形成不同类区。

表 1　多层网络结构与分类区域的关系

网络结构	分类区域类型	对异或(XOR)分类	对咬合状区域的分类	分类区域一般形状
无隐层 	由超平面分成两个区域			
单隐层 	开凸区域或闭凸区域			
两个隐层 	任意形状(但与节点数之间的关系复杂)			

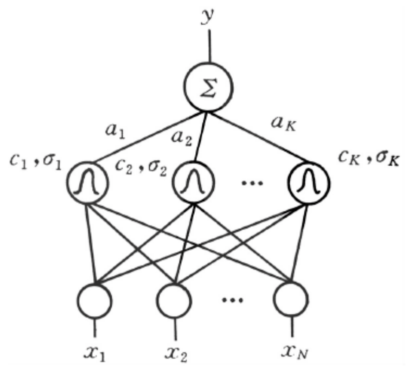


图 2　用径向基函数的多层前向网络

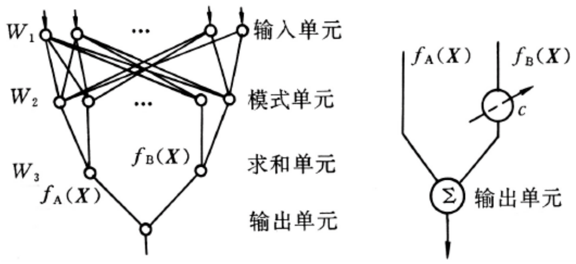
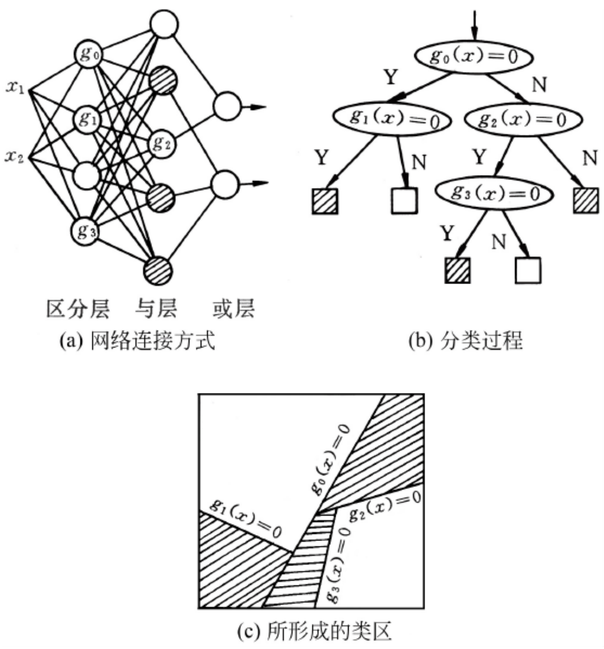


图 3　概率神经网络



(a) 网络连接方式　(b) 分类过程　(c) 所形成的类区

图 4　用神经网络实现二值决策树

深度学习网络(deep learning)是一种多层神经网络。该模型被证明可以对高维结构化数据有效建模。

神经网络也被用于聚类分析,其中最常用的是自组织映射和相应的学习算法。

人工神经网络还可用于特征压缩。对于一个具有 N 个输入的单个神经元,如果输入 $\boldsymbol{X}=(x_1,x_2,\cdots,x_N)^{\mathrm{T}}$ 是均值 $E[\boldsymbol{X}]=0$、协方差阵 $\boldsymbol{R}=E[\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}]$ 的随机向量,设输出为 y,权系数 $\boldsymbol{W}=(w_1,w_2,\cdots,w_N)^{\mathrm{T}}$,可以证明如果权向量按下式修改:

$$\Delta\boldsymbol{W}=\mu y[\boldsymbol{X}-\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{W}]$$

其中 μ 为步长,则 $\boldsymbol{W}$ 将收敛于对应 $\boldsymbol{R}$ 最大特征值的特征向量,据这一方法可以实现主成分分析,从而可用于特征的压缩。

有一种基于知识的分类器,它显式地利用知识和推理规则把样本分类(相当于一个用于模式识别的**专家系统**),它也可以用神经网络实现,这种方法的缺点是学习过程复杂。如果把两种方法结合起来构成用神经网络组成的混合型分类器,则既可以方便地利用知识又具有很好的学习能力。

模糊推理与神经网络结合具有一系列优点。在模式识别方面,用神经网络可实现基于模糊推理的分类器和模糊聚类算法,也可以实现如前所述的基于模糊知识和映射的混合型模式识别系统。

参考文献

1. Zurada J M. Introduction to artificial systems. West Publishing Company, 1992

2. 阎平凡,黄端旭. 人工神经网络——模型,分析与应用. 合肥:安徽教育出版社,1993

3. Lippmann R P. Pattern clasification using neural networks. IEEE Communication Magazine, 1989, 27(11):47-64

(阎平凡)

rengong zhineng

人工智能(artificial intelligence, AI)　模拟与解释人类智能的一门学科,其中智能可理解为是一种能力,如洞察、学习、思考、理解和推理的能力。

1. 人工智能学科的任务和早期研究

人工智能的研究有两个相辅相成的任务,一是发展具有类似生物(人类)智能的计算系统(智能信息处理或智能系统),并以此来解决困难的实际问题;二是借助计算机模拟并解释生物(人类)的智能(认知科学)。尽管有的研究者将后者划归为人工智能学科,但是,还是有很多研究者将人工智能的研究限制在前者的理论和方法上,这样,人工智能也就自然成为计算机科学的一个分支。

在 20 世纪 40—50 年代期间,为了实现"制造具有某种人类智慧的聪明机器"的理想,人们至少提出了三种途径:

一是对环境适应。1948 年,Wiener 提出控制论,强调智能表现为"对变化的外界环境的适应",其关键是反馈。这个思想后被 Ashiby 发展为以微分方程定性理论为基础的控制理论,基于此,Ashiby 写出了《大脑设计》的著作。这个思想在以后人工智能的发展中并没有起到主导作用。直到 1991 年,Brooks 提出临场人工智能时,这个思想才进入人工智能的研究范畴。

二是神经信息处理。鉴于智能来自神经活动的信息加工过程,根据神经科学的研究结果,建立智能模型是一个合理的选择。这类研究与 1943 年 McCulloch 与 Pitts 的神经元信息加工方式的模型,以及 1948 年 Hebb 的学习机制的研究有密切关系。建立模型的方法一般以统计学(优化)为基础。

三是认知科学。在人类行为层次建立智能模型,其最重要的研究是 Turing 实验,即,一种根据行为判别机器是否具有智能的准则。1951 年,Shannon 首先设计并实现了第一个计算机下棋程序。相对于神经信息加工的智能研究,它不介意行为产生的原因,且模型一般以符号推理为基础。

在"模拟与解释人类智能行为"研究的半个世纪中,"对环境适应"一直没有成为主流,除了"机器昆虫"之外,在理论与方法上,似乎并没有本质性的影响。基于认知科学(复杂信息处理)与基于神经信息加工两种实现智能行为的研究,交替引领着这类研究。

1956 年 Dartmonth 会议上,五位计算机科学家 McCarthy,Simon,Newell,Minsky 与 Shannon 使用"人工智能"这个术语来概括基于认知科学符号信息加工的研究,自此,人工智能成为计算机科学一个分支。

2. 人工智能学科研究的主要内容和 20 世纪 60—90 年代研究进展

人工智能学科的主要研究内容包括:**知识表示**,**自动推理**和**搜索**方法,**机器学习**和**知识获取**,知识处理系统,**自然语言处理**,**计算机视觉**,**智能机器人**,自动程序设计等方面。

直到 20 世纪 60 年代末,人工智能一直处于奠基阶段,但是,这个时期发展的理念、理论和方法,至今还是人工智能的基础。1969 年,Minsky 和 Papert 合写了《感知机》一书,表面上看,这本书指出了感知机算法存在的问题,但是,对人工智能来说,它是一个分水岭。正是这本书,使得基于符号处理的智能研究进入了"以知识为核心"的黄金十年。人工智能的理念得到工业界与社会的认可,以知识获取和使用为内容的知识工程研究成为人工智能研究的核心课题。

人工智能发展的奠基阶段有着许多重要的研究结果,它们至今还影响着人工智能甚至整个计算机科学的发展,例如,McCarthy 的符号处理语言原型,这是计算机科学符号处理的计算基础;Simon 夫妇的心理学实验,表明人类的问题求解是一个搜索的过程,其效率取决于启发式函数,Nilsson 的启发式搜索算法 A^*,以及 Robinson 的归结原理。它大大推动了自动推理的发展。

表示与知识表示是人工智能的基本问题之一。表示是描述实际问题的一个框架(或称为基,或称为编码方式),知识表示则是,在这个框架下,对某个具体问题为真的描述,即,这个问题的模型。对表示而言,常用的表示语言有:逻辑、产生式、语义网络和框架,以及多项式等。

对人工智能来说,知识表示有两类研究。一类与**专家系统**密切相关,即,如何有效表述专家知识,称为经验知识。另一类则是将研究集中在如何表示常识知识。常识知识是指人们直觉的、日常使用的那些非专业性知识,它们不一定在任何情况下都正确,例如,"鸟会飞"是常识知识,大多数鸟都会飞,但也有众多例外,如"鸵鸟不会飞"。但是,由于这类知识往往使得问题求解时的搜索空间大大减小,因此,对复杂问题的求解常起决定性的作用。如何表示和使用常识,自然为人们所关注。

推理是知识的使用过程,由于有多种知识表示方法,相应地有多种推理方法。一般地说,推理过程大致可以分为演绎推理和非演绎推理。谓词逻辑是演绎推理的基础,**语义网络表示**和**框架表示**下的继承性推理是非演绎性的。演绎推理与定理证明有密切关系。同时,由于知识处理的需要,随之提出了多种非演绎的推理方法,如连接机制推理、类比推理、**基于案例的推理**和受限推理等。在推理方面,我国学者也做出了令人瞩目的成果。吴文俊教授提出的平面几何定理的机器证明方法,证明了大量相当困难的平面几何定理,受到国际学术界的重视,被称为**吴方法**。吴方法的重要性有两点:一是在表示上,与基于逻辑的方法相比较,吴方法将问题限制在可以表示为代数方程的范围,而不是逻辑表示那样普适;二是在推理上,吴方法是非线性代数方程的一种消元法,其意义已超出了人工智能的范畴。

搜索是人工智能问题求解方式。搜索策略决定

着问题求解的一个推理步中知识被使用的优先顺序。可分为无信息导引的盲目搜索和利用经验知识导引的**启发式搜索**。启发式知识常由启发式函数来表示，启发式知识利用得越充分，求解问题的搜索空间就越小，A^*、AO^*是启发式搜索算法的代表。

整个20世纪60年代，人工智能的研究基本奠定了符号处理的理论与技术。进入70年代，人工智能进入新的时期，即，进入知识处理的时期，也可以称为专家系统或更为广泛的**知识工程**时期。

1970年左右，Feigenbaum发表了第一个专家系统DENDRAL，用于分析化学分子结构。1977年，Feigenbaum将这类研究总结为知识工程。人们逐步认识到领域专家通过实践积累起来的知识的重要性，并总结出建造专家系统及开发环境的一系列原则。这期间出现了成千上万的专家系统，涉及上百个应用领域，如医学、地质学、分子生物学、化学分析、自动程序设计、计算机辅助教育和智能控制等。这些成果使人们看到了人工智能研究的实际意义，推动了整个人工智能的发展。

与此同时，在广泛的应用过程中，人们也发现专家系统获取专家知识的困难性以及系统本身的脆弱性。这些弱点使得耗资巨大的CYC知识库计划（大百科全书计划）没有获得预期的效果。尽管如此，在研究专家系统期间发展的基本方法，例如，**知识库**方法与各种**推理机**等已成为各个应用领域不可缺少的技术。

1997年，IBM公司的“深蓝”下棋系统打败了国际象棋大师卡斯帕罗夫，这可能是机器第一次打败世界上超一流棋手的事件，这是“知识库加**启发式搜索**”最重要的成果，这在某种程度上实现了Turing的梦想。

1986年，Rumelhart等人出版了PDP（Parallel Distributed Processing）报告，其中一章中给出了一个非线性的多层感知机算法，称为反向传播算法（Backpropagation，BP）。这个算法使Minsky在“感知机”一书中的提问——XOR问题可以收敛，由此引起轰动，进而，感知机的研究被扩展为**人工神经网络**。在以后的研究中，人们为了与基于符号机制的智能相区分，将人工神经网络的研究称为智能的连接机制。尽管其科学基础是神经科学，但是，其实现是以统计学为基础。当时，这两种机制产生了激烈的交锋，争论的本质是：智能行为应该采用结构描述方法还是统计描述方法。

另外，在20世纪80年代后期，计算机处理符号的目标已经完成，而单纯基于符号的人工智能理论与方法却遇到障碍，这导致“模拟与解释人类智能行为”的研究呈现出百花齐放的态势。人工智能开始接受来自其他学科的思想和研究成果，由此，出现了一些新的方法。例如，Holland提出的基于自然选择的遗传进化模型，使用类似基因结构的表示法来描述问题，利用类似生物遗传的方法来发现新的规则；Brooks找回Wiener与Ashiby等对环境适应的控制论方法，使用机器昆虫的行为展现智能，这是一种不使用符号表示的人工智能研究，也有人称其为“临场人工智能”，等等。

3. 人工智能学科近年的研究进展

20世纪90年代以来，尽管人工智能学科的研究遇到的严重的困难，但还是出现了一系列至今对计算科学有重要影响的研究结果，其中最为重要的是，1991年，Minsky出版了他的著作，《Society of Mind（思维的社会）》。这本著作关于“智慧由一些小的无智慧的独立功能单元组合后产生”的建议，以及创造的新术语“Agent”，已引起理论与应用研究者的关注。Agent已成为计算机科学很多领域广泛使用的概念，据此还发展成为知识表示的方法论——本体论（Ontology）。

进入20世纪90年代，统计机器学习逐步开始引领人工智能研究的主流。这时，对人工智能十分重要的表示与推理的研究，由于基于优化的学习算法大多数采用给定基函数，因此，其表示变得单一，且由此导致推理成为计算模型函数的简单问题，表示与推理在统计机器学习中失去了研究价值。

统计机器学习在以后的20年间，并没有沿着BP的非线性算法的路线发展，反之，回归线性感知机是其特点。在Valiant的概率近似正确（Probability Approximation Correct，PAC）学习理论意义下，Vapnik提出了支持向量机（Support Vector Machine，SVM）。尽管统计方法在这个时期占据了主流地位，但是，人工智能的研究者并没有忘记“智能”的含义，因此，在这个时期，发展了大量不同的学习方式，这些方式大多来自对人类学习的研究，例如，流形学习、主动学习、**集成学习**、多示例学习等。

这个时期，这类研究分为两个不同的研究路线：一是以PAC为基础，强调学习过程可以基于有限样本，并使得对误差的分析以$1-\delta$概率成立。这个路线的最重要的贡献是强调建立模型的算法应该在线性空间设计，即，强调返回线性感知机，这是对BP算法设计的反叛，由此，导致至今还是重要的研究课

题——核函数。

另一个有趣的路线是遵循传统统计学理念,根据热力学的"系综(Ensemble)"、神经科学的"集群(Ensemble)",以及统计学的重采样(Resampling)等原理发展了现在称为"集成学习(Ensemble Learning)"的方法。其本质是,对实际问题随机采样并建立模型,采样次数进行多次,由此获得多个模型,然后,在这些模型张成的空间上建立实际问题的模型。在统计学上证明,如果采样次数趋于无穷,由此建立的模型的均方差与一次采样建立的模型的均方差相等,这就是已被广泛应用于各个不同领域的 Bootstrap 原理。与此同时,1991 年人们证明了弱可学习定理,由此发展了算法 Boosting。它与上述随机采样的区别:一是对给定样本集合的采样,二是下一次采样尽量包含上一次采样建立的模型不能准确描述的样本,因此,Boosting 需要建立在 PAC 基础上。

此期间最广为人们所喜爱的研究结果是"最大间距(Margin)"算法,其误差界依赖样本集合两个闭凸集之间的距离(Margin),即,距离越大,泛化性能越好。由于这个原理的几何解释十分清晰,由此设计的算法简单易懂,因此,被很多研究者所喜欢。

目前,统计机器学习遇到艰难的问题:数字网络、分子生物学和金融等领域迫切需要解决大变量集合下大数据的分析与处理问题,其关键是,大变量集合的统计学与海量样本集合。后者,涉及快速算法,而前者不仅是对计算机科学的挑战,而且更多地是对统计学的挑战。其困难在于,由于空间巨大,使得将一个样本考虑为这个空间上一个点的做法,永远无法获得足够多或足够充分的样本。

经过 20 年的研究,一方面,大变量集合统计学遇到根本的困难,另一方面,网络、生物学和金融等不同领域的迫切需求,使得人们不得不寻找更有效的方法。人们想起 20 年前的概率图模型。

概率图模型的本质就是在变量集合中找出所有条件独立的变量对,并构成部分关联、其他不相关联的非平凡的图结构。在变量之间的连线上,附贴了一个概率分布。

由于采用这样的结构表示,其问题求解的过程将需要推理,换句话说,任何对这个模型的查询,需要从这个图结构上计算与查询有关变量的边缘分布,这是一个 NP 完全的问题。目前已有大量近似解法,但是,这是一个需要进一步解决的重要问题。

与计算机科学和技术的其他分支一样,人工智能的研究同样需要社会的检验。但是,人工智能与其他分支的区别是,它关注当前还没有答案的那些问题。在我们的世界中,这样的问题占据的大多数。这就是在计算意义下,人工智能存在的最重要的理由。在当前的社会中,具有上述性质的领域多不胜举,最典型且经常遇到的有:自然语言处理,图像分析与处理,信息检索,机器人,生物信息处理,以及金融分析与预测等,其中,有些领域已经成为人工智能的一部分,例如,自然语言处理,机器人;另外一部分,成为人工智能的一部分只是时间问题(例如,生物信息的分析与处理,"Science"有专辑作过说明)。

人工智能的研究已有 50 多年(甚至更长)的历史,发展是曲折的,从制造具有智能的机器的梦想来看,相距甚远;从计算机应用的角度来看,其成果甚丰。不夸张地说,它已经成为计算机应用发展的原始动力之一。

参考文献

1. 陆汝钤. 人工智能(下). 北京:科学出版社,1996

2. Rumelhart D E, et al. Parallel distributed processing: Explorations in the microstructure of cognition (Volume 1: Foundations). Cambridge, MA: The MIT Press, 1986

3. Vapnik V. The nature of statistical learning theory. New York: Springessr-Verlag, 1995

(石纯一　王珏)

rengongzhineng luoji

人工智能逻辑(logic for artificial intelligence) **人工智能**在处理非规范知识时所用的非经典逻辑。人工智能研究的是如何让计算机学会人的思维规律,而逻辑是思维的规范。更现实地说,**知识表示**和**自动推理**是人工智能系统必须具备的功能,为此人工智能需要借助各种数学工具,其中首要的是逻辑。**一阶逻辑**在人工智能中被用作重要的和基本的知识**表示架构**,一阶逻辑中的确定子句的归结方法是人工智能的确定性推理的核心机制。但是,这种"经典"的逻辑方法只能处理"规范的"知识,而"非规范"知识大量地出现在常识中。如何在计算机中处理人类的常识及常识推理,并建造具有常识的智能系统,成为人工智能基础研究的一个核心问题和关键技术,它也是影响人工智能进一步发展的基本问题。对于人工智能必须面对的各种"非规范"知识,我们必须寻找和研究新的逻辑方法。所谓知识的非规范性,是指知识内涵的难处理性,包

括知识的不确定性(模糊性、随机性和不精确性),或知识的不完整性(内容不完整性和结构不完整性),或知识的不协调性(含矛盾,带噪声和含冗余等),或知识非恒常性(时变性和启发式知识)。处理"非规范"知识的逻辑通常是"非经典逻辑"。

处理不确定性知识的逻辑

现实问题领域中的知识往往是不确定的,处理不确定性知识的形式系统中最重要和最具影响力的是**模糊逻辑**。模糊逻辑是一种一阶逻辑,其真值取0~1之间的实数。扩充真值域的本质在于将二值判断推广为多值判断,也就是将逻辑命题的真实性以及逻辑推理的正确性进行程度化。最简单的办法就是用[0,1]中的实数来表示这种程度。模糊逻辑在数据处理方面取得了一定成绩,在模糊控制方面取得了很大的成功。但是,对于像因特网数据处理这样一些现代技术和现代生活提出的问题,模糊逻辑缺少有效的办法,需要研究新的理论和方法。

主观逻辑采用与模糊逻辑不同的方式处理不确定性:它认为具有模糊性的是人们对命题的主观信仰。主观逻辑被用于建立网格环境下的信任模型。

处理不完整性知识的逻辑

常识推理往往是在信息不完整的情况下进行的,为了有效地进行推理经常采用如下思维方式:"如果没有证据反驳A,那就承认A。"在这种推理模式下,增加新的信息可能会导致原来在该模式下可推出的结论失效,从而出现非单调性。自20世纪70年代以来,相继出现了各种逻辑系统用于形式描述常识推理中的非单调性,它们统称为**非单调逻辑**。

具有代表性的非单调逻辑理论包括:R. Reiter的**缺省逻辑**、J. McCarthy的**限定逻辑**、D. McDermott与J. Doyle的模态非单调系统、R. C. Moore的自认知逻辑以及由D. M. Gabbay、D. Makinson、Y. Shoham、S. Kraus、D. Lehmann及M. Magidor等人发展起来的非单调后承理论。

非单调逻辑的理论研究已经取得了比较丰硕的成果。然而,非单调推理系统的成功应用案例却很稀少。其主要原因之一在于系统的表达能力和有效实现的矛盾。通过对语言表达能力进行限制,可以实现具有较高效率的非单调推理系统,例如,采用了"失败即否定"机制的逻辑程序设计语言就可以比较高效地实现一些类型的非单调推理。在此基础上出现了实用性较强的**应答集程序设计**。

处理不协调性知识的逻辑

从理论上说,一个一阶逻辑系统只要含有一个矛盾(非协调),就会使任何命题在此系统中为真,从而使该系统成为一个平凡的系统。由于矛盾无处不在,我们当然希望在含有矛盾的情况下仍然能够进行有意义的推理,为此需要相应的逻辑支持。**超协调逻辑**就是一类非协调但不平凡的理论。目前超协调逻辑已有众多的逻辑系统,如超协调命题演算C_n通过减弱演绎能力达到形式化超协调性的目的;超协调多值逻辑通过指派至少两个真值定义真理,从语义上可直接具有超协调性;准经典逻辑定义了两种可满足关系:弱满足关系和强满足关系,前者在结论上用来容忍矛盾而后者在前提上使消解规则成立。

传统上,超协调逻辑属哲学逻辑范畴,目前在计算机科学和人工智能中具有重要的应用。而在计算机科学和人工智能背景中超协调逻辑也得到了进一步的发展。

用于特定知识领域的逻辑

用于特定知识领域的逻辑有:用于本体和语义网研究的**描述逻辑**,用于智能体理论研究的BDI逻辑,用于时空对象描述和处理的时空逻辑,以及**情景演算**,事件演算,流演算等。上面列举的特定知识领域不一定是相应逻辑的唯一来源,但肯定是它们目前最重要的应用领域。

知识表达的一种有效方法为:先定义应用领域("世界")的有关概念(术语),然后用这些概念表达该领域中对象和个体的性质(世界描述)。描述逻辑为这种知识表达方法提供理论基础(包括形式语义和推理机制)。

描述逻辑的主要应用领域是本体论和语义Web(参见**语义Web的逻辑基础**)语义Web体系架构中的核心层——本体层是以描述逻辑为逻辑基础的。采用描述逻辑作为语义Web的本体语言的主要原因是:描述逻辑能够为语义Web提供对其至关重要的高质量的本体,而且本体的构建、整合和进化很大程度上依赖于描述逻辑的明确语义和强大推理功能。

近年来智能体理论和技术在人工智能中的重要性日益显著。解释对智能体理性行为产生影响的心理因素(信念、愿望、意图等)的基本框架称为(belief-desire-intention(BDI))结构。一个BDI模型(参见**智能体的BDI模型**)包含三种基本成分:①信念(Belief)是智能体对世界的认知,包含描述环境特性的数据和描述自身功能的数据;②愿望(Desire)是智能体希望达到的状态或希望保持的

状态的集合；③意图(Intention)是当前智能体正要实现的目标。

BDI 逻辑是研究 BDI 结构中诸因素的特征和相互关系的逻辑的统称。研究工作表明，BDI 模型的形式化不是一件容易的事。因为信念，愿望，意图这些意识概念实质上是模态算子，通常的 BDI 逻辑是一种多模态逻辑，把信念、愿望和意图模型化为可能世界语义下的正规模态算子。正规模态逻辑存在“逻辑全知”问题，由此带来一系列问题使 BDI 模型的形式化工作遇到困难，同时也激发了大量的理论探索工作。

时空逻辑是描述时空对象及其相互关系的逻辑语言，由时态逻辑和**空间逻辑**发展而来。近二十年，时空逻辑研究在人工智能、空间数据库、图像处理等领域得到很大发展。

时态逻辑是时间表示的计算模型。空间逻辑是描述空间几何实体及其相互关系的逻辑语言，可以是一阶逻辑、一阶逻辑的片段或高阶逻辑，所涉及的空间可以是任何一种几何空间。现实世界中需要同时表示对象的时间和空间属性，随着对空间逻辑和时态逻辑研究的不断深入，同时表示对象的时间和空间属性及其关系已成为可能。广义上讲，描述时间、空间属性及其相互关系的逻辑形式化方法都属于时空逻辑的研究范畴。

时空逻辑选择一种时态逻辑和一种空间逻辑，把它们融合成一个时空混合体。构造时空逻辑主要有两种方法：一是从语法出发由已有的空间逻辑和时态逻辑构造时空逻辑，给出时空逻辑的公理及解释；另一种方法是从语义出发，对时态和空间逻辑的语义进行扩展，主要以命题、一阶和模态逻辑为基础。二者都以构造表达能力较强且计算复杂性较低的时空逻辑为目标。

人工智能逻辑的研究虽然已经有很长的历史并且已经取得相当丰富的成果，但是，它还难以解决非规范知识处理中的一些重要问题。最为困难的问题可能是表达能力与逻辑推理复杂性之间的折中与权衡。另外，现有的人工智能逻辑往往只能处理单一的非规范性，能统一处理多种非规范性的逻辑是一个重大的挑战。

参考文献

1. 陆汝钤. 人工智能(下). 北京：科学出版社，2002

2. Van Harmelen F, Lifschitz V, Porter B, eds. Handbook of knowledge representation, elsevier, 2008

(刘椿年)

renji jiaohu jishu

人机交互技术(human-computer interaction technology, HCI technology) 依据对人类交互行为(包括人与人，人与计算设备之间的交互行为等)的理解，使计算设备能够以友善、自然和直观的方式与用户进行交互或为用户提供信息服务的技术。

随着计算技术、网络和传感器技术的发展，当前的计算设备已经通过各种方式联网并与多种传感器集成，其应用也渗透到人类生活的各个方面。相应地，计算设备的使用人员也由计算机专业人员扩展到非专业人员。简单、易学、易用是用户对这些复杂计算设备的基本要求，人机交互技术是当前普适计算、以人为中心的计算等新计算模式的重要基础，逐渐成为计算机应用研究与技术的一个重要方向。

人机交互是一个典型的交叉学科，其技术的形成是多学科综合作用的结果。计算、网络与传感技术的发展是人机交互技术进步的基础，**计算机图形学**、**多媒体技术**、信息处理、**计算机视觉**等学科发展与人因工程学、工业工程、认知心理学、社会心理学等学科的发展，共同促进了人机交互研究的进展，推动着人机交互技术的广泛应用。计算机图形学的发展与鼠标的出现，形成了新的人机交互技术，图形用户界面(GUI, graphic user interface)使人们能够用直观的直接操纵的方式操作计算机，从而推动了个人计算机的产生、发展和普及。多媒体计算和智能接口技术的发展又进一步使得有可能以人类惯有的方式进行人机交互。计算机操作系统和相应的开发软件一般都包含有“用户接口管理系统”和“用户接口开发工具”，对系统的个性化使用和应用程序用户接口的开发提供了支持。随着各种传感器与计算设备的集成，计算设备的输入方式从简单的键盘、鼠标变化成为多种传感器。计算机视觉等信息处理方法与技术也成为人机交互技术中不可缺少的一部分。第二次世界大战时期对设备易操作性的要求引起了人们对人因工程的关注，在人机交互技术的研究中也要考虑用户如何方便地使用计算机设备。人类工程学研究用户工作时生理特性的影响并着重研究环

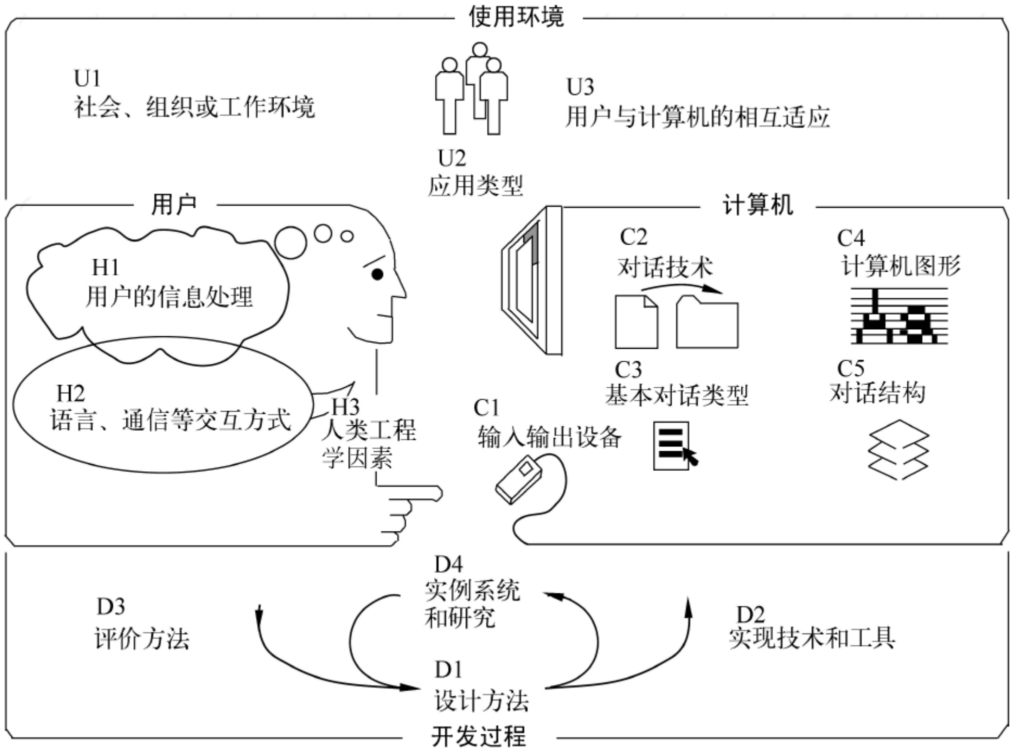


图 1　人机交互技术主要的研究内容

（图片引自参考文献 3）

境和系统对用户的影响。工业工程学来源于 20 世纪初提高工业产量的需求，人机交互技术也需要研究如何让计算机应用到更多的工作中以提高生产效率。认知心理学和社会心理学有助于人机交互技术的进步，如研究用户学习某种计算机系统的能力、用户获得的这种能力在其他计算机系统中的适应程度以及用户在这些系统中的工作效率等。

人机交互技术主要的研究内容有人机系统执行任务的综合性能，人机之间的交互系统结构，人类学习使用计算机的能力，交互方法与系统的设计、实现及其评价。它们的相互关系如图 1 所示。计算设备总是要和一定的社会、组织或者工作环境（U1）相联系，在各种环境中用户希望使用的应用类型（U2）也不尽相同，工作的过程意味着用户和计算机相互适应的过程（U3），也就是需要用户学习计算设备的使用方法，需要系统为用户定制功能等。除了研究使用环境外，从用户方面而言，人机交互技术需要了解用户的信息处理过程（H1）、用户交流的方式（H2）以及用户的生理特征（H3）等。从计算设备的角度来说，人机交互技术包括人机之间的输入输出设备（C1）（尤其是近年来传感器技术的飞速发展，使得现代计算系统的输入方法产生了很大的变化）、人机之间的对话技术（C2）、对话的基本类型（C3）；进一步而言，对话技术可以采用多媒体技术、计算机图形技术和虚拟现实技术等（C4），复杂的对话技术需要考虑对话的体系结构（C5）方面的因素，如多用户协同界面、多任务对话模型等。最后，在开发阶段，人机交互技术需要考虑集成人机对话的设计方法（D1）、实现技术和开发工具（D2）、对系统的评价方法（D3）以及实例研究（D4）等，开发阶段的每一个过程是相互关联、彼此影响的。

人机交互领域国际最有影响的组织为 ACM SIGCHI（Special Interest Group on Computer Human Interaction），最有影响的国际会议有由该组织举办的 CHI，最有影响力的期刊是 ACM Transactions on Computer-Human Interaction（TOCHI）。

参考文献

1. 徐光祐，陶霖密，史元春，张翔. 普适计算模式下的人机交互. 计算机学报，2007，30（7）：1041-1053

2. Faulkner C. The essence of human-computer interaction. Prentice Hall，1998

3. Dix A J，Finlay J E，Abowd G D，Beale R. Human-computer interaction. 2nd ed. Prentice Hall，1998

4. http://sigchi.acm.org

（陶霖密　史元春）

renji jiaohu xitong

人机交互系统(human-computer interaction system) 支持人和计算机系统直接进行交互通信的系统,其主要功能是完成人机之间的信息传递以提高计算机系统的友善性和效率。

分时系统出现后,用户可以在各自的终端上使用计算机,例如,从终端打入命令和输入数据,并从终端上得到计算机系统输出的各种信息,这就是早期的人机交互系统。这种人机交互系统的输入输出设备是具有输入输出功能的控制打字机,相应的软件是命令解释系统。

随着计算机系统(包括输入输出设备)功能的增强,人机交互系统也有了较大发展。新型输入输出设备,例如,CRT交互终端设备、鼠标器、声音输入和合成设备、图像扫描仪、文字识别设备等的出现,使人机交互系统从用文字进行交互通信发展到可以通过图形、图像、声音等进行交互通信。近来多媒体技术的出现,使人机之间有可能按照人的自然和习惯的方式,通过多种载体或媒体完成信息的交换和处理。

人机交互系统要能很好地实现用户与计算机之间的人机交互,必须考虑三个元素:人的因素、交互设备及实现人机交互的软件。

其中,人的因素是指用户操作模型。要根据用户的类型,固有的特点,设计好的用户操作模型,使人机交互系统满足用户的使用要求。交互设备构成了人机交互系统进行人机对话的基础。它包括数字和字母输入输出设备、图形和图像输入输出设备,以及声音、触感等专用输入输出设备。人机交互软件是人机交互系统的核心,它向用户提供各种交互功能,以满足系统预定的要求。它和所有软件一样可分为系统软件和应用软件。

在系统软件方面,许多分时操作系统均采用命令语言的对话方式向用户提供操作界面,这类操作系统如UNIX,VMS,DOS等。一些高级语言的解释程序(如BASIC,LISP,PROLOG)采用交互式解释执行,而高级语言的编译程序(如Turbo Pascal,Turbo C)则采用编辑、编译、调试等交互式集成程序设计环境,这类语言工具十分便于用户编程和调试。在数据库管理系统中通常也采用交互式数据库查询语言,有的用命令语言(如SQL),也有的用填表方式(如QBE)。在数量众多的软件工具中,已经广泛使用全屏幕正文编辑程序、排错程序,电子表格软件、多窗口系统等交互式软件工具。系统软件中还包括一批可用于辅助生成人机界面的软件工具或环境,应用系统的人机界面可以在它们的基础上开发,或用它们进行辅助开发。多窗口系统、用户界面管理系统(UIMS)等就是这样的工具。交互式图形系统也是这类支持软件之一。

在应用软件方面,交互式人机界面已成为其主要部分之一,并成为衡量应用软件功能强弱的一个重要指标。在人机交互应用系统中,开发人机界面的部分占了相当大的工作量。为了提高人机界面软件的生产率和可重用性,一个重要的发展趋势是将人机界面与应用系统中的功能部分分离出来,并研制自动或辅助生成人机界面的软件工具。由于应用领域的广泛性,不同应用领域的人机交互方式可能迥然不同。

根据人机交互方式的演变,人机交互系统可以大致分为命令语言交互系统,选单驱动交互系统,直接操纵交互系统,多媒体交互系统等。

命令语言交互系统 命令语言是用以请求系统服务并传送系统回答的可执行语言。它由一组命令组成,每一命令又由命令名和若干命令参数构成。命令语言交互系统负责获取用户命令,分析命令的语法、语义结构,实际执行命令赋予的功能,并把系统回答传送给用户,从而让用户通过键入命令来控制和操纵计算机系统的运行。

选单驱动交互系统 选单是若干可供用户选择的系统功能表。选单驱动交互系统是通过选择选单项来执行系统功能。选单的内容及组织结构体现了系统的功能及其按层次的分解与组织。选单驱动交互系统鼓励用户在选单引导下遍历系统功能。

直接操纵交互系统 直接操纵交互系统借助图形并通过模拟人对客观世界中实体的操作来完成人机交互并使操作过程和操作效果可视化。例如把表示文件的图符移到表示废纸箱的图符中则比喻对该文件实行删除操作。直接操纵使用户可以用现实生活中认识和处理问题的方式和习惯来解决人机交互问题,所以很受用户欢迎并已广泛用于许多人机交互系统中。

在直接操纵交互系统中,广泛使用了WIMP(窗口、图符、选单、指点器)式图形用户界面技术和面向对象方法。

多媒体交互系统 多媒体交互系统采用文本、图形、图像、声音、视频等多种媒体来表示、储存和处理信息,通过人们耳闻、口述、目睹、手触等多种方式与计算机进行交互,使人机交互更加简单、自然、友善、

一致。

人机交互系统的研究内容主要有：

（1）人机交互系统模型的建立与分析　它研究如何把人的认知模型包含到计算机系统设计中去，建立描述人机交互系统的工作原理、组织结构和人机交互活动过程的人机交互模型。常用的人机交互模型有：基于语言描述的结构化分层模型、基于描述时间和逻辑序列的控制模型、基于应用任务的任务分析模型及面向对象模型等。

（2）人机交互系统工作方式和设计原理　它依据交互输入输出设备和计算机软件技术的发展，研究适合用户需要的人机交互方式、制定和总结各种交互方式的人机界面的设计原理与准则，来指导界面的设计和用户的选用。

（3）人机交互系统的设计方法　它研究如何设计和开发界面的屏幕外观形式、确定用户和系统交互方式，并把用户操作处理成对系统功能的控制。常用的设计方法有：使用程序设计语言（如C，C ++）、使用界面工具箱（如 X 窗口下的 X-toolkit 和 Widget 部件集）、使用专门的界面描述语言（如 Motif 标准中的 UIL 语言）、使用直接操纵方式等。

（4）人机交互系统的评估　它研究如何评价人机交互系统的功能，制定各种评价准则。主要评价性能有：使用的难易程度、学习的难易程度、开发的难易程度、系统的复杂程度、操作速度等。常用的评价方法有：随机分析方法、概率统计方法等。

参考文献

1. 程景云，倪亦泉. 人机界面设计方法与开发工具. 北京：电子工业出版社，1994

2. 董士海. 计算机用户界面及其工具. 北京：科学出版社，1994

（程景云　倪亦泉）

renlian donghua

人脸动画（facial animation）　通过计算机合成人脸表情的动画技术。它实际上涉及人脸造型和人脸运动控制两个问题。这方面的研究内容包括人脸几何造型、由照片合成人脸、语音驱动表情动画、表情捕捉和重现、脸部纹理映射、脸部形态变换、脸部细节模拟、表情库和表情合成等。在脸部表情合成方面，一种简单的方法是数字化仪将人脸的各种表情输入到计算机中，然后用这些表情的线性组合来产生新的脸部表情。该方法的缺点是缺乏灵活性，不能模拟表情的细微变化，并且与表情库有很大关系。1987 年，Waters 提出了一个基于 Facial Action Coding System 的脸部表情动画模拟方法。该方法由一个参数肌肉模型组成，人的脸用多边形网格来表示，并用肌肉向量来控制人脸的变形。它的特点在于可用一定数量的参数对模型的特征肌肉进行控制，并且不针对特定的脸部拓扑结构。Waters 用该方法生成了高兴、恐惧、愤怒、厌恶、惊奇等逼真的表情，Waters 的算法已成为模拟脸部表情动画的核心算法之一。

参考文献

1. Kerlow IV. The art of 3D computer animation and effects. 4th ed. Wiley Publishing, 2009

2. Parke F I, Waters K. Computer facial animation. 2nd ed. Natick, MA: A K Peters, 2008

（金小刚）

renlian shibie

人脸识别（face recognition）　计算机利用人的脸部图像自动识别其身份的过程和技术。计算机对输入的图像或视频，首先判断其中是否存在人脸，如果存在，则进一步给出人脸的位置、大小以及各个主要面部器官的信息，并依据这些信息，进一步提取脸像中所包含的身份特征信息，与人脸库中的已知人脸进行对比，从而识别其身份。

脸像是人类视觉中最为普遍的模式，人脸所反映的视觉信息在人与人的交流、交往中有重要的作用和意义。人脸识别的研究从 20 世纪 60 年代末开始，由于其广泛的应用领域，其研究一直备受瞩目，依托于图像理解、**模式识别**、**计算机视觉**和神经网络等技术，其研究取得了很大进展，并逐渐推向应用领域。

人脸识别领域比较著名的国际研究机构有美国麻省理工学院媒体实验室及人工智能实验室、南加州大学（USC）、CMU 卡内基-梅隆机器人研究及交互系统实验室、马里兰大学（UMD）等。

国内对人脸识别的研究从 20 世纪 90 年代中后期开始，许多研究机构在人脸识别研究领域进行了许多有意义的尝试，在大规模人脸识别领域已经取得突破性进展，并投入了实际应用。

完整的人脸识别技术至少要包括两个主要环节。首先要在输入的图像或图像序列中找到人脸的位置，并把人脸从背景中分离出来，这是人脸检测的主要研究内容。所发展出来的方法主要有基于模板匹配、基于颜色信息、基于特征脸匹配、基于 AdaBoost 的实时人脸检测方法等。其次，根据分离后的

人脸图像进行特征的提取与识别,即人脸识别环节。人脸特征主要包括人脸的全局特征、几何特征、各个器官的特征等,因此为适应这些特征也产生了不同的识别算法。

(1) 基于几何特征和基于模板匹配的方法　这两种方法主要在人脸识别研究的初期使用。基于面部结构几何特征的人脸识别是最直观,也是最传统的方法,它对人脸的描述非常紧凑,但存在特征提取困难、易受头部姿态变化影响等缺点。基于模板匹配的方法则具有特征提取简单的优点,其准确度也较高,但该方法容易受到光照和姿态的影响。因此,这两种方法通常都需要与其他算法结合,才能得到比较好的效果。

(2) 特征脸识别方法　基于特征脸(eigenface)识别的方法由麻省理工学院 M. Turk 和 A. Pentland 于 1991 年提出,是目前最流行的人脸识别算法。该方法通过主分量分析(PCA)将原始脸像分解成“平均脸”和一系列互相正交的“特征脸”,并基于此进行识别。图 1 给出了平均脸和特征脸的分解示意。

该方法具有简单有效的特点,但它对输入的人脸图像的归一化要求较高,性能易受光照和姿态变化的影响。目前研究者对其进行了各种改进和扩展,尝试了基于特征脸和各种后端分类器相结合的方法,如二阶 eigenface、线性判别分析以及双子空间分析方法等。

(3) 基于融合特征的方法　近年来,研究人员逐渐认识到,一种有效的人脸识别算法,必须能够充分挖掘人脸不同方面的特征信息,也即要有效利用人脸的形状拓扑结构、局部灰度和全局灰度分布等多种特征。因此,出现了融合上述特征信息的一些方法。

基于 Gabor 小波变换和弹性图匹配的方法是一种人脸形状拓扑结构特征和局部灰度特征的算法。另一种融合多种面部特征的方法是基于可变形统计模型的方法,该方法综合利用人脸形状信息(主动形状模型 ASM)、局部灰度分布特征和全局灰度分布特征(主动外观模型 AAM)。上述方法具有自动提取形状特征或精确定位特征点的功能,因而具有较高的识别精度,但是存在复杂度高、实现复杂的缺点。

(4) 其他方法　此外,局部特征分析(LFA)、隐马尔可夫模型(HMM)、奇异值分解(SVD)、独立元分析(ICA)等方法也在人脸识别中得到了应用。

人脸识别技术具有广泛的应用前景。在各种安全领域,如智能门禁、视频监控等都具有重要的应用价值。在游戏娱乐领域,人脸识别也有一些有趣有益的应用,比如能够识别主人身份的智能玩具、具有真实脸像的虚拟游戏等。另外,图像捕捉设备正成为个人计算机的标准外设,因此人脸识别也必将成为未来人机和谐交互的一个重要手段。

必须看到,尽管人脸识别技术已经取得了长足的进展,但是在应用过程中所遇到的一些实际问题,比如光照条件的改变,用户姿态、表情、发型、胡须的变化,真实人脸与照片的辨别,甚至不同摄像机参数的不一致等,都严重影响到识别的性能。必须对这些实际问题进行系统的、有针对性的研究。

参考文献

1. Turk M, Pentland A. Eigenfaces for recognition. J Cognitive Neuroscience, 1993, 3: 71-86
2. Martinez A M, Kak A C. PCA versus LDA. IEEE Trans PAMI, 2001, 23(2): 228-233

(蔡莲红　吴志勇)

renti donghua

人体动画(human body animation)　利用计算机进行人体造型和行为模拟的**计算机动画**技术。它综合运用**计算机图形学**、解剖学、生理学、生物力学、机器人学、物理学、**人工智能**等学科领域的理论、方法和技术,生成尽可能逼真的角色形状和运动。人体动画的研究内容包括角色造型、角色运动形变、运动控制、表情动画、毛发动画、服装动画、自然语言驱动的动画、运动捕获、运动重现和自主运动等。

人体是由关节构成的,关节动画是人体运动控制的核心。驱动关节链结构运动的方式主要有两种:运动学方法和动力学方法。运动学方法仅考虑物体的位置、速度和加速度等运动参数,这种运动与物体的质量、产生该运动的隐含力无关。而在动力学方法中,运动参数完全由动力学方程所决定。关

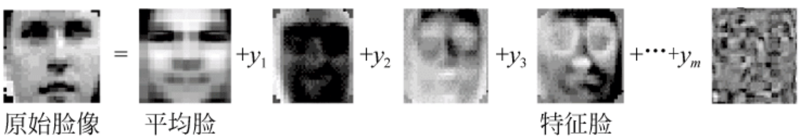


图 1　特征脸分解示意图

节结构由一系列刚体链在关节处连接而成，其中链结构的自由端称为末端影响器。虽然在机器人学中有平移关节和旋转关节两种类型，但在计算机动画中只有旋转关节。在数学上，描述关节链的常用方式为DH表示，它的特点是将坐标系附在每一个链上来描述一个链相对于其他相邻链的运动，两个相邻链坐标系的齐次坐标变换矩阵称为A矩阵。在正运动学中，所有关节向量的值由动画师显式给出，末端影响器的位置由关节间变换的复合得到。而逆运动学是一种目标导向的运动指定方法，简称IK。在逆运动学方法中，动画师指定末端影响器的位置，系统求解关节向量的值。在动力学方法中，物体的运动是由力和力矩驱动的，由于计算所得的运动是符合物理规律的，故能生成更复杂和逼真的运动。动力学方法又可分为正动力学方法和逆动力学方法。在正动力学方法中，给定作用于物体的力和力矩，计算各关节的位移、速度和加速度；而在逆动力学方法中，力和力矩是通过给定的运动学参数反求得到的。

目前一种非常流行和实用的人体动画技术是运动捕获和运动重现。与通过交互方式设置关节动画的方式不同，运动捕获采用软、硬件系统自动记录表演者的真实运动信息，并把动作结果施加到一个虚拟的人或动物上。捕获的人体关节运动数据通常采用BioVision公司的BVH格式来储存。运动捕获主要包括光学运动捕获系统、磁性运动捕获系统和机械式运动捕获系统。运动捕获的数据是表演者真实运动的映像，但表演者和虚拟人的关节结构数据(如身高、腿长、手臂长度等)不一定完全相同。若不加修改地直接把运动捕获的数据直接应用到一个不同尺寸的虚拟人上，就有可能出现运动不协调、双脚离地等不真实的运动。运动重现可以把一个角色的动画赋给另一个具有相同关节结构但具有不同关节长度的角色，并能保持原有动画的质量。

人体动画是现代高科技电影的核心技术之一，它使得用计算机生成真实影视产品成为可能。导演可使用计算机指挥、调度各个人造角色、舞台场景、灯光和摄像机，在一个虚拟世界里产生出真实的影视佳作。人体动画在影视业、娱乐业、广告业、体育训练、舞蹈教学、医学等多个领域得到了广泛的应用。

参考文献

1. Parent R. Computer animation: algorithms and techniques. 2nd ed. San Francisco, CA: Morgan Kaufmann, 2007

2. Kerlow IV. The art of 3D computer animation and effects. 4th ed. Wiley Publishing, 2009

（全小刚）

renwu diaodu chengxu

任务调度程序(task scheduler)　并发程序设计系统中用于调度和分派处理器的程序。又称进程调度、处理器调度或低级调度程序。

在并发程序设计系统中，在同一时刻可能有多个**进程**同时竞争处理器，这就需要按任务调度算法，把处理器占用权交给就绪队列中的一个进程，以便让它执行。任务调度的主要功能有：记录进程状态和维护进程状态队列、分派处理器和回收处理器。

在单处理器系统中，任务调度算法主要解决如何按一定的策略把空闲处理器分派给一个就绪进程。评价一个任务调度算法的主要依据有：系统吞吐率和响应时间。常用的任务调度算法有：

(1) 优先数调度　优先数调度算法是基于设置的进程优先数，把处理器分派给就绪队列中优先数最高的就绪进程。

设置进程优先数的方法可分为：静态和动态优先数法。静态设置方法是指在创建一个进程时，按照该进程的进程类型、资源需求等因素由系统指定一个优先数，而且，在进程整个生命周期该优先数保持不变。这种方法易于实现、开销较小，但不够灵活、效率较低。常用于**批处理**操作系统。动态设置方法是指进程生命期内，由系统根据进程占用处理器的时间长短、进程任务的缓急程度、资源的均衡使用等因素动态地改变进程优先数。这样能获得更精确的调度和更佳的调度性能。

(2) 轮转法调度　时间片轮转调度算法是轮流调度所有就绪进程，即每隔一个时间片，依次从处于就绪状态的诸进程中选一个运行。该算法主要用于分时系统。

轮转法调度进程的关键一是要利用闹钟，定时发出时钟中断，以调度另一就绪进程运行。二是决定时间片大小，时间片过大，退化为优先数法，难以实现轮转执行。时间片过小，导致调度和分派进程过于频繁，增加系统开销。时间片大小的确定应综合考虑多种因素，如系统响应时间、联机终端个数、处理器处理速度及系统的其他处理能力。

根据时间片的大小、就绪队列的个数，可把轮转

法调度分成简单循环轮转调度、变长时间片轮转调度、多级反馈队列轮转调度。

(3) 分级调度　又称反馈队列或多级队列调度。该调度算法的主要思想是将就绪进程列入多个不同级别的就绪进程队列。任务调度每次从高级就绪进程队列中选取可占用处理器的进程,只有在选不到时,才从较低级就绪进程队列中选取。进程分级可事先规定,例如,使用外围设备频繁者属于高级;分时系统中的终端用户进程定为高级。进程分级可动态进行,例如,凡运行超越时间片后,就进入低级就绪队列,以后赋给较长的时间片;凡启动磁盘、磁带而成为等待的进程,在等待结束后就进入中级就绪队列等。

随着分布式系统的发展,分布式任务调度算法成为研究的热点。其研究内容之一是设计将一组相互协作的任务分配到一组处理器上运行的分配算法。分布式任务调度算法的评价标准与单处理器系统相比,增加了对负载的要求,即要求系统中各处理器具有较为平衡的负载,避免出现个别处理器特别繁忙,而其他处理器非常空闲的情形。

参考文献

孙钟秀,等. 操作系统教程. 北京: 高等教育出版社,1989

(郑宇华)

rongchi wangluo

容迟网络(delay tolerant networks, DTN)

具有间歇连通特性,应用能够容忍一定延迟的一类无线网络。由于为减少成本节点稀疏部署或是为节省能量节点进行休眠以及无线通信距离有限、节点移动和通信噪声等原因,使得网络中的无线链路时常断开,通信源节点和目标节点之间瞬时没有一条完整的端到端通信路径。数据包在传输的过程中可能被阻塞在中间节点上,当新的无线链路建立时才能继续传输,数据包的传输延迟比较大。这样的网络一般使用在对延迟要求不是很严格的网络系统中,要求传输协议和应用程序能够容忍一定的通信延迟,因此称为容迟网络,有时也被称为容断网络(disruption tolerant networking, DTN)。

容迟网络概念最初来源于美国国防部高级研究计划局(DARPA)的星际互联网计划(IPN)。星际互联网的目标是在星球之间建立一个互联网,地球上的 Internet 只是其中的一个子网。由于星球按照一定的轨迹运动,它们之间的距离发生变化导致星际链路反复地连接和断开,整个星际互联网在瞬间来看不是全连通。为了解决不同子网间的间歇性连通和实现子网间的通信,互联子网的网关采用“存储-携带-转发”(store-carry-forward)的通信模式,当存在到达目的网络的可用链路时就转发消息,否则将消息存储在本地等待可用链路。2002 年 K. Fall 利用 IPN 的思想设计了地面无线网络(terrestrial wireless networks),提出了容迟网络的概念。

容迟网络的网络拓扑在多数情况是一个随时间动态变化的非连通图,通信源节点与目的节点间瞬间不一定存在端到端路径,传统网络的“存储-转发”模式不适应于这种间断网络。容迟网络采用“存储-携带-转发”的通信模式。在移动应用场景中,源节点和转发节点携带着数据包一起移动,直到出现新的邻居时,由转发算法判断其是否是下一跳转发节点,数据包被发送给下一跳节点,经过多个转发节点的中继,消息最终传递到目标节点。其实质是利用节点的移动,在一个非连通的网络中经过逐跳转发实现端到端的通信。在一些非移动的应用场景中,如无线传感器网络,为了节省能量,节点采用休眠机制,在数据包的传输路径上,部分节点可能处于休眠状态,整条传输路径在瞬间也不连通,数据包需要等待一下跳节点醒来时才能被继续传输。虽然容迟网络拓扑的瞬时快照可能不连通,但是把连续的瞬时快照叠加在一起时将变得连通。从应用的角度,容迟网络中的应用程序对传输延迟的要求不是非常严格,希望在延迟和网络成本之间取得折中。

典型的容迟网络有星际互联网络、移动 Ad Hoc 网络(MANET)、车载 Ad Hoc 网络(VANET)。容迟网络有多个类似的概念,如断连移动 Ad Hoc 网络(intermittently connected mobile Ad Hoc networks)、机会网络(opportunistic networks)等,这些概念都是从网络通信特性的角度来阐述的,如机会网络强调利用节点移动的机会性通信传输数据。

参考文献

1. Fall K. A delay-tolerant network architecture for challenged Internets. In: SIGCOMM. Karlsruhe: ACM, 2003: 27-34

2. 熊永平, 孙利民, 牛建伟,等. 机会网络. 软件学报, 2009, 20: 124-137

(孙利民)

rongcuo jisuan

容错计算(fault-tolerant computing)　计算机系统具有容错能力的计算。

容错是指容许系统在运行过程中发生一定的差错或故障时仍能保持正常工作而不影响正确结果的一种性能。具有容错功能的系统称为容错系统，该系统在出现某些硬件故障或软件差错时能自行检测和排除故障或差错，以保证完成预期的任务，并得到正确结果，整个过程不需要用户的干预，因而对用户来说是透明的。容错与可靠性（参见**计算机系统可靠性**）有着密切的关系，可靠性是系统性能评价中的主要指标之一。容错计算则是提高计算机系统可靠性的有效手段和措施，通过容错措施的实施可以大大地提高可靠性。但是容错不是提高可靠性的唯一措施，例如一种与容错的概念恰恰相反的措施称为避错技术也是提高可靠性的一种有效措施。

一个完整的容错计算系统至少包括以下 10 个组成部分：故障限制、故障检测、故障屏蔽、复执、故障诊断、重配置、回复、重启动、修复、重构等。简要分述如下：

（1）故障限制　一旦系统发生故障，要求限制该故障的影响范围。将由于故障引起的“污染”限定在系统的最小范围内而不使其扩散。故障限制可以用故障检测、一致性检查等方式完成，它既可以由硬件、也可以由软件来实现。

（2）故障检测　故障检测技术一般分为两大类：脱机检测和联机检测。在进行脱机检测时，被检系统不能正常运行，例如在系统上运行诊断程序时必须暂停执行系统的应用程序。而联机检测则提供实时检测的功能，此时检测与正常应用程序可同时运行。

（3）故障屏蔽　是将已发生的故障隐蔽起来，不使暴露，使用户察觉不到故障的发生。故障屏蔽与故障检测正相反，前者是掩盖由某些故障引起的错误信息，后者则要激活使其产生错误信息，以便检测到故障的存在。故障屏蔽多采用冗余技术来实现，如多数裁决器。

（4）复执　在计算机硬件系统中的故障可以分为两类：永久型和暂时型。永久型故障一旦发生不会自行消失；而暂时型故障又分为两种，间歇型和瞬时型，前者的发生为断续且不定周期的，而后者为偶发一次性的。复执技术正是针对瞬时型故障的偶发性特点而设计的，当因瞬时型故障而产生错误时，重复多次执行相同的指令可能会因故障消失而得到正确的结果。因此，复执是一种代价较小、效果较好的容错计算措施。

（5）故障诊断　故障诊断包括两部分：故障检测和故障定位。故障检测已如上述，故障定位则是对故障检测的进一步加强，即不仅要检测出故障的存在而且要确定故障发生的部位。在容错计算系统中，必须能确定故障部位以进行纠正或修改，从而获得最终的正确结果。

（6）重配置　当故障被检测并定位以后，就需要对故障部件做出相应的处理，这称为系统的重配置。重配置方法有两种：一是用备份部件代替故障部件（如动态冗余技术等）；二是直接将故障部件从系统中分离出来，此时系统仍能正常工作，但功能已有所减弱，称为降级使用或称为正常降级。

（7）回复　系统发生故障并经重配置后，将运算回复到发生故障前的某一断点，这种恢复的方法称为回退法，它是利用备份文件、校验点以及流水日记账等方法记录其运行轨迹，然后回退到适当的断点，重新恢复所需的数据等。这里要注意的是必须确定故障的潜伏期，若发生的故障具有一段潜伏期，则在回退时必须回退到该故障的潜伏期前，以免再次遇到未被检测出来的故障而造成的错误。

（8）重启动　由于发生了故障，系统中的信息可能多处遭破坏甚至无法恢复原来的信息，因此需要进行重启动。重启动一般分为三类，热启动、温启动和冷启动。热启动是指在断点前的所有信息均未遭破坏直接从断点开始的启动；温启动是指在断点前有某些信息被破坏，但这些已破坏的信息可以通过备份或记录来恢复，恢复后可以自断点进行启动；冷启动则是指所有信息都必须重新装入后才能进行的启动。

（9）修复　确定故障部件后必须进行修复，修复方式有脱机修复和联机修复两种。脱机修复是指故障部件与系统分离情况下的修理；联机修复是指系统自动切除故障部件换上备份部件，继续工作，或自动切换故障部件，使系统降级继续运行。

（10）重构　当所有故障部件被替换并重配置以后，系统应回复到初始的基本结构状态。例如三模冗余电路，在发现有一个模块发生故障后，应及时切换下来，并将备份模块投入运行，重新构成三模冗余电路，以保证系统的可靠性。

容错技术可以分为 3 类，即故障检测技术、屏蔽冗余技术和动态冗余技术。

（1）故障检测虽不直接提供容错功能，但它可在发生故障时发出警告。由于实现这种技术的成本较低且较容易，因此常被应用于微、小型计算机设计中。这些系统往往将故障检测技术与联机检测相结

合,检测虽是检错,而不是容错,但它是容错计算的前提,没有检错功能,也就无法进行容错计算。

(2) 屏蔽冗余又称静态冗余,它与故障检测功能相反,对故障或错误的出现不做警告处理,容许错误存在而保持系统的正常运行。例如纠错编码和多数裁决冗余电路等,都是屏蔽冗余技术的具体实施方案,这些技术在容错计算系统中使用十分广泛。

(3) 动态冗余是在静态冗余电路的基础上根据系统在运行过程中发生的故障随时改变系统结构的冗余技术,动态冗余技术实际上是静态冗余、联机故障检测及重配置等几项技术的结合,它在静态冗余电路工作的同时对系统做检测并将测得的故障模块自动切换或将系统降级使用,因此该技术具有很强的实时性和容错功能,是高可靠系统容错设计的重要技术之一。但是实现这种技术的成本较高。对系统的运行速度也有一定的影响。

近年来流行的高可用系统是一种成本较低的容错系统,主要用于避免软件错误造成的停机,两台或多台机器互为备用机,平时各执行不同的任务。但一台机器出错时应用程序迅速切换到另一台备用机上运行。另外,随着计算系统虚拟化技术的发展,采用虚拟化技术,用软件的方法构建多个互为备份的虚拟机也成为实现高可用系统,并同时降低成本和能耗的常见方法。

容错系统是20世纪70年代初开始在国际上发展起来的,至今已有40多年的历史。随着微电子技术及计算技术的不断的迅猛发展,对容错计算的理论和技术将会提出更高的要求,同时也会得到更广泛的应用。用容错计算的策略提高可靠性是一个十分积极而又实际的思想,容错计算的研究将会有很大的发展。

参考文献

1. IEEE Standard Computer Dictionary, 610. 12, 1990

2. Sieviork D P, Swarz R S. The theory and practice of reliable system design. Digital Press, 1974

3. Johnson B W. Design and analysis of fault-tolerant digital systems. Addison-Wesley Publishing Company Inc., 1989

4. Jalote P. Fault tolerance in distributed systems. Prentice Hall, 1994

(邵志远 徐拾义)

rongcuo jisuanji

容错计算机(fault-tolerant computer) 在计算机系统的某些硬件部件发生故障或软件产生错误时,系统仍能保持继续运行并提供相应服务的计算机。它被称为具有容忍故障能力的计算机。容错计算机的主要设计目标是提高计算机系统的可靠性、可用性、安全性和可信性等性能。提高计算机可靠性的方法可以分为两大类:一类称为排错(fault removal)技术,主要是通过严格的筛选和测试等方法以尽量使用高可靠性的元器件达到减少系统发生故障的目标;另一类则是容错技术,主要是运用冗余技术来抵消或者克服由于系统元件故障而引起的影响。所谓冗余技术,简单地说,是在正常系统运行所需的部件的基础上加上一定数量的附加信息、时间或后备硬件、后备软件的技术。由于一个计算机系统从研发、生产制造到投入运行的整个生命周期中,发生或遇到故障是必然的,因此,从应对故障的角度分析,排错技术只是一种消极的应付方法,而容错计算技术是积极应对故障的方法,具有广泛的应用前景。冗余技术是容错计算机所采用的最主要的技术基础之一,大致上可以分为下列几种类型:①硬件冗余 以检测或屏蔽故障为目的而添加一定硬件设备;②软件冗余 为了检测或屏蔽软件中的错误而添加一些可以检测故障但在正常运行时未必需要的软件;③信息冗余 在实现正常功能所需的信息以外,再附加一些补充信息来验证原信息正确性,例如纠错编码就是信息冗余的一种形式;④时间冗余 附加一定的系统运行时间来重复执行和检测系统的功能,这些附加的时间主要是用在故障检测或故障屏蔽上。

常用的硬件冗余是硬件的重复和叠加使用。硬件冗余一般可以分为三种类型:静态冗余(或称为被动冗余)、动态冗余(或称为主动冗余)和混合冗余。静态冗余的思想是将可能出现的故障屏蔽起来,使其不影响系统运行的结果。静态冗余主要是依靠表决机制来屏蔽发生的故障,它不需要故障检测也不必进行系统的重新配置等就可以获得容错的效果。静态冗余技术中典型的是**三模冗余**技术(TMR)。TMR的基本概念是使用3个完全相同的模块执行相同的任务,然后由1个多数表决器对3个模块的输出进行比较和表决以确定整个系统的输出。多数表决器的表决原则是三中取二,也就是说三模冗余系统可以容许有1个模块发生故障而不至于影响到整个系统运行的正确性。三模冗余技术的关键是多数表决器本身的可靠性问题,提高多数表决器可靠性的方法有多种,其中最常用的方法是多

数表决器本身也使用三模冗余,即利用 3 个独立的多数表决器,每个多数表决器分别接受来自 3 个模块的输出作为它的输入,然后再分别输出。这种系统通常被称为带三重多数表决器的三模冗余系统。除了三模冗余系统外,还有多于三模冗余的技术,称为 N 模冗余技术。动态冗余技术与静态冗余技术相反,它是通过故障检测、故障定位及故障恢复等手段达到容错的目的。因而,动态冗余技术并不是防止故障引发错误,而是将错误暴露出来,从而去纠正错误。动态冗余技术中最典型的方法是构造带有比较器的双工系统。该方法使用 2 个相同功能的硬件系统,且同时完成相同的任务,然后对它们的结果做比较。如果 1 个双工系统只有 1 个比较器,则只能检测到有无故障,却无法确定哪一个模块出了故障。所以在这样的系统中还必须增加一定的措施才能确定故障的位置。动态冗余技术除了上述方法以外,还有使用诸如热备份模块、把关定时器(又称为看门狗定时器)等较为常用的方法。硬件冗余的第三种类型则称为混合冗余。这种技术是将动态和静态冗余相结合的冗余技术,以取二者之长处。它在前置部分是使用静态冗余的故障屏蔽技术,使系统免受某些可以屏蔽故障的影响,而对那些无法屏蔽的故障则在后置部分采用动态冗余中的故障检测、故障定位和故障恢复等技术来处理,并且对系统可以作重新配置。因此,混合冗余的效果要大大优于单独使用动态和静态冗余技术。然而,由于混合冗余既要有静态冗余的屏蔽功能,又要满足动态冗余技术中的各种检测、定位等功能,它的附加硬件和软件等的开销是相当大的,所以混合冗余的成本很高,仅在对可靠性要求极高的场合中采用。混合冗余的方法也有多种,例如,带热备份的 N 模冗余技术、自清洗冗余技术、筛选模块冗余技术等。

软件冗余是利用冗余的软件来检测系统中故障的方法。利用冗余软件进行故障检测的方法很多,常用的有一致性检查、能力检查和多版本程序设计等技术。一致性检查是对某一运行结果先做一定的预测,然后在程序运行中和运行后将其结果与预测结果做比较。若实际结果在期望值的范围内,则认为正常,若实际结果超越了期望值的范围,则认为有故障。能力检查是用检查程序去检查系统中各个部件应有的能力,例如用程序来读写某一个存储单元,以检查该单元的存储和读写能力,又如用一组特定的数据去检查运算逻辑部件,以判断该部件能否进行正常的运算等。多版本程序设计技术则是对一个相同的任务(或算法)用不同的方法进行独立的程序设计,然后对不同版本的程序运行后,将得到的结果进行同步比较(这里的同步概念与硬件同步概念有所区别,前者在同步时间上并不是严格要求的,它可以在一个时间区间段得到同步结果就可以了),若所有版本运行的结果相同,则认为无故障,否则,就认为有故障存在。值得注意的是,这种方法实际上是来自于硬件冗余技术中的 N 模冗余的思想。多版本程序设计不仅能检查硬件故障,也可以检查软件本身的故障,因此,在软件容错技术中经常使用。

信息冗余是一种将附加信息添加到相关的数据上从而达到故障检测、故障屏蔽和容错目的的理论和方法。信息冗余最好的例子就是检错编码和纠错编码。这是将冗余的信息加到一个数据字上使每一个数据字变为一个新的带有冗余信息的字。这种冗余信息的添加方法是按照一组预定的编码规则进行的,符合编码规则的冗余信息字称为码字(codeword),而不符合编码规则的字则称为非码字(noncodeword)。按冗余信息的添加规则在信息字上添加冗余信息位的过程称为**编码**;反之,将已编码的码字恢复成原来信息字的过程则称为**译码**。一般来说,经过编码的码字只是全部编码的子集,另一部分则是非码字。当系统出现故障时,可能会将码字变成非码字,于是在译码过程中会将造成非码字的故障检测出来,这就是检错码的基本思想。至于纠错码则不仅可以将信息位中的错误检测出来,而且还能自动地将发生故障的非码字纠正或恢复成原来的码字信息以达到纠错的效果。由此可见,信息冗余技术的主要任务在于应用附加的信息位通过编码和译码技术来提高信息流的可靠性。编码技术中最简单的检错码之一是奇偶校验码,奇偶校验的基本思想是在二进制的信息字上附加一位冗余位,称为校验位,使得该码字(这里的码字是信息位加上冗余位而形成的信息字)中所含有 1 的个数为偶数或为奇数。如果编码规则规定码字中含有的 1 的个数为偶数,则称这种奇偶校验为偶校验;如果规定码字中 1 的个数为奇数,则称这种校验为奇校验。由于奇偶校验码简单实用,便于硬件实现。因而在计算机系统中被广泛采用。但是奇偶校验码存在一定的缺陷,例如它不能检测偶数个同时发生的故障,因而在它的基础上又发展了一系列扩展的奇偶校验码,例如分段奇偶校验码、分字节奇偶校验码等。事实上,纠错编码理论是一门较为独立的具有深厚数学基础

的学科,应用十分广泛。在当今数字信息时代,这门学科正在不断的发展并得到更多的应用。可以直接应用于数字系统编码的主要有线性编码和非线性编码两大类,而每一大类又可以分为分组码和卷积码。例如,数字系统中常用的海明编码(Hamming code)就是线性分组码中最典型的代表。这个编码不仅能够检测出二位错误,而且还能纠正一位错误,将非码字改正为码字,因而被称为纠一检二码,已经被广泛采用。

时间冗余是以时间(即降低系统运行速度)为代价以减少硬件冗余和信息冗余的开销来达到提高系统可靠性的目的。在某些实际应用中,由于使用硬件冗余和信息冗余时,发现其成本、功耗、重量等开销过高或者系统体积过大,而运行速度(或运行时间)方面却并不是一个重要因素的情况下,则可以使用时间冗余技术来替代需要使用硬件冗余和信息冗余技术才能完成的功能,以达到容错计算的目的。时间冗余的基本概念是重复多次执行相同的工作,或称为重复执行,简称复执,以达到检测故障的目的。实现时间冗余的方法很多,但是其基本思想不外乎是对相同的计算任务重复执行多次,然后将每次的运行结果存放起来并进行对比。若每次的结果相同则认为无故障;若存在不同的结果则说明检测到了故障。不过,这种方法往往只能检测到瞬时型故障而不宜检测永久型的故障,这是因为瞬时型故障会使各次运行产生不同的结果。若不仅要检测瞬时型故障,而且还要检测固定故障等永久型故障,则单靠时间冗余是有困难的。因此,在系统中还必须附加少量的冗余硬件。时间冗余与硬件冗余的结合,既能检测瞬时型故障,又能检测永久型故障。

上面介绍的硬件冗余、软件冗余,信息冗余和时间冗余等冗余技术是保证数字系统可靠性,获得容错计算功能的基本措施和主要方法。在实际应用中,上述4种冗余技术经常是结合起来使用的。将这些冗余技术融合在一个计算机系统中,可以成为一个具有容错计算的冗余系统。

一般说来,一个较为完整的容错计算机系统,在处理运行中出现的故障时,大体上需要具有以下十大主要功能:

(1)故障检测　这是处理故障的基础,因为容错系统是指容许故障存在时能够保持正常运行的系统,所以,系统必须具有故障检测的能力。故障检测的方法很多,如上述的奇偶校验就是检测故障的一种方法。故障检测一般分为两类:联机检测和脱机检测。前者提供了实时检测的能力,这种检测工作与系统的正常工作同时进行;后者进行检测时,系统必须停止正常工作。

(2)故障屏蔽　是指一旦发现故障,立即将出现的故障屏蔽起来,使系统免受故障的影响。

(3)故障限制　即在发生故障以后,需要将故障所产生的影响限制在一定范围之内,防止已发生的故障影响到系统的其他部分。

(4)重复执行　这是一种检测瞬时型故障的有效措施,它可以提高计算机抗瞬时型故障干扰的能力。

(5)故障诊断　在故障检测的基础上,对发现的故障进行定位,这对以后进行的故障部件修复和重配置等工作有很重要的意义。

(6)系统重配置　若故障一旦被检测并定位,则系统应有将发生故障的元件或部件自动替换下来的能力,或者具有将故障部件与其他部分隔离开来的能力。当故障部件被替换下来后,系统中虽然可能缺少了这一部件,但仍应能保持正常运行,只是系统的运行速度或能力有所下降。这一功能称为系统的降级使用。

(7)系统恢复　当检测到故障,必要时在系统重配置后即可消除故障引发的错误。这时,系统应能返回到出现故障断点前的情况开始继续运行。这个过程称为系统恢复过程。

(8)系统重启动　如果系统由于出现过多的故障而造成大量的错误,以致无法恢复破坏了的信息造成系统恢复失效时,就不能再使用上述系统恢复过程,而必须自动进入系统重新启动阶段。重新启动分为热启动和冷启动,前者是在部分信息遭到破坏但还有一部分可以利用的情况时使用,而后者则是在几乎所有信息均遭破坏的情况下使用,即系统在初始化的情况下启动工作。

(9)系统修复　凡是已确定有故障的部件必须进行修复,修复分为脱机修复和联机修复两种。若要修复的部件卸下后对系统影响很大,或者修复这些部件时系统必定会停机,就使用脱机修复。联机修复通常是指系统能自动启用备份部件替代有故障部件,并保持系统继续运行,然后再修复切换下来的故障部件。

(10)系统重组合　当上述各步完成后,系统必须重新组合,以便完全恢复到原来的正常运行状态。

容错计算机主要应用于工业生产、医疗、航空、航天、军事、公安、交通、金融、通信等对计算机的可

靠性要求很高的部门和场合。然而,以计算机为代表的“硅”应用不断渗入人类社会的事实,却是一把双刃剑:一方面“硅”给人类社会带来无限广阔的应用天地为人类造福;另一方面,一旦以“硅”为核心的数字系统出现无法控制的故障和失效情况,则可能会给人们带来重大损失甚至造成严重的灾难。这样的事实,可以从近年来发生在国内外的一系列重大、特大安全生产隐患事故得到证实。这些事故中的大部分原因,都无不同计算机控制和信号系统的可靠性有着十分密切甚至是直接的关系。这些恶性事故的发生,虽然给人们带来损失和灾难,却也使人们进一步提高了对可靠和可信系统重要性的认识。而提高系统可靠性和可信性最佳也是最直接的方法之一就是在关键部门和场合,必须应用具有容错计算功能的系统。

从长远来看,虽然开发和研制高可靠性容错计算机的成本远高于研制具有相同计算能力的一般计算机系统,但是,随着微电子技术的迅速发展以及在高端技术应用中迫切需求的推动下,容错计算和容错计算机理论和技术正在向深度和广度发展,尤其是在硬件和软件容错理论、测试算法、诊断技术和自适应容错计算机等方面的研究和开发,均迫切需要继续深入研究。

参考文献

1. Barry W Johnson. Design and analysis of fault-tolerant digital systems. Addison Wesley Publishing Company Inc., 1989

2. Siewiorek D D, Swarz R S. The theory and practice of reliable system design. Digital Equipment Corporation, 1982

3. 徐拾义. 可信计算系统设计和分析. 北京:清华大学出版社,2006

4. 徐拾义. 容错计算系统. 武汉:武汉大学出版社,2010

(徐拾义)

rongcuo sheji

容错设计(fault-tolerant design) 当计算机系统中的某些部件发生故障后,使系统功能能够继续保持正常或者保持部分正常地运行(即平缓降级)而避免全面失效的设计。在工程中也称为“失效安全”的设计。对于计算机系统而言,当硬件或软件发生故障时,采用容错设计的系统可能在吞吐量或响应速度等性能指标上有所降低,但系统的执行线索不会因故障而中止或被修改,并且执行结果也不包含系统中故障所引起的差错。狭义的也指当故障产生并引发错误时避免系统失效的设计方法。它与避错设计及失效安全设计共同构成广义的容错设计概念。

容错设计的基本手段是采用冗余。当某一个部件出现故障时,它所承担的工作必须由与其配对的无故障部件来接替。如果这些部件仅用来提高系统的可靠性,而不影响系统的计算性能,则称它们为冗余。冗余有四种基本形式,即硬件冗余、软件冗余、信息冗余和时间冗余。按冗余资源的使用方式,冗余技术又可分为被动冗余、主动冗余和混合冗余。采用何种冗余取决于容错系统在具体应用环境中的需求。实现容错设计主要有 4 个方面的内容,即①故障、错误、失效等不希望事件的检测;②损坏估价;③不希望事件的恢复及不希望事件的处理;④继续服务。

注意区分可靠性设计和容错设计这两个概念。其区别在于:**可靠性设计**侧重于对产品的可靠性进行预计、分配、技术设计、评定等工作,解决的主要问题是如何从设计入手来解决产品的可靠性,以改善对复杂系统的各个零部件可靠度的要求,其核心是可靠度的分配;容错设计侧重于引入额外的资源来提高整体的可靠度,解决的主要问题是如何利用冗余手段达到指定的可靠性指标从而满足应用的需求,其核心是冗余技术的运用。

参考文献

1. 胡谋. 计算机容错技术. 北京:中国铁道出版社, 1995

2. 杨士元. 数字系统的故障诊断与容错设计. 2 版. 北京:清华大学出版社, 2000

(尚利宏)

rongzai xitong

容灾系统(disaster recovery system) 为完成容灾功能所设计的计算机软、硬件系统。容灾(disaster recovery)是指在某种自然或人为灾难所导致的信息系统损坏后,为了重新获得对计算机硬件、存储数据、网络通信等资源的正常访问,以便重启和恢复关键业务的正常运行,而需要完成的一系列步骤、过程和策略。

容灾系统最早出现于 20 世纪 70 年代中期,那时的计算机系统是面向批量处理的**大型计算机**,一旦由于某种原因而停机,为恢复系统运行可能要花费较多的时间,从而影响正常业务的运转。这使得人们开始意识到计算机系统容灾的重要性,并衍生

出一个以提供备份计算系统为主要业务的产业。80年代到90年代,由于**个人计算机**的发展带动了计算设备的普及,计算机产业对国民经济的影响不断提升,容灾产业发展迅猛。而进入21世纪后,随着网络技术的发展,计算机系统需要持续不断地为客户提供服务,即使短暂的业务中断也变得越来越不可容忍。

容灾系统所针对的灾难种类包括:①自然灾难　例如地震、火灾、水灾、气象等;②人为灾难　例如错误操作、黑客攻击、病毒发作等;③技术灾难　例如设备失效、软件错误、通信中断、电力失效等。当这些灾难发生时,有时造成的不仅是信息系统的经济损失,某些关键系统的损坏甚至可能造成社会的不安定。因此容灾系统已经成为信息社会的基本保障性系统。

不同的容灾系统所针对的错误类型可能也不同,这些错误按照其严重程度分,主要可分为4种,包括:①临时性错误　比如网络中断所引起的服务中断;②崩溃性错误　主要是内存及处理器中程序状态的丢失;③媒介错误　指存储于磁盘等非易失性存储器中的数据丢失;④区域性错误　指所有在某一地理区域(比如一整栋楼甚至一个城市)内的计算设备完全损坏。

通常一个容灾系统在设计时所关注的目标有两个,一是恢复点目标(RPO),实际是指容灾系统进行备份时的时间间隔是多少,它决定了在发生灾难后可能丢失的数据多少;二是恢复时间目标(RTO),是指系统在遇到灾难后,需要多长时间能够恢复正常运行。针对不同应用场景对RTO和RPO的要求,1992年Anaheim的M028会议提出的SHARE78是目前国际通用的异地远程恢复标准,它将信息系统的灾难恢复能力划分为七个层次:本地数据备份与恢复、批量存取访问方式、批量存取访问方式+热备份地点、电子链接、工作状态的备份地点、双重在线存储、零数据丢失。SHARE 78对每一个层次需要满足的要求都做了详细的说明。

容灾系统主要技术实现方式包括:

(1)周期性备份技术　周期性将数据进行备份是最为传统的一种容灾备份方式。根据备份所使用的载体不同,它又可以分为磁带备份和磁盘备份。磁带备份技术虽然比较古老,但近年来却也发展成了另外一种备份形态,就是**虚拟带库**(VTL)。一个虚拟带库就像一个物理磁带库一样,但提供了更大的配置灵活性,而不用像物理磁带库一样,只能与备份数据进行1对1标记。虚拟带库目前主要的技术进步还在于利用了重复数据删除技术,这样提高了磁带的存储密度以及备份数据的读写效率。磁盘备份技术则逐渐在取代传统磁带备份的地位,主要原因有两个:一是磁盘的存取速度比较快,二是磁盘可以做随机存取,有利于进行存取优化。

(2)远程镜像技术　远程数据镜像通常用于**存储区域网**(SAN)环境,它利用SAN的底层网络远距离连接能力和统一存储的特点,在异地保存一份与本地数据相同的拷贝,保证了重要数据在遭受区域物理灾难后的可用性,是现代容灾系统的重要组成部分。按照远程镜像中数据写协议的不同实现,它可分为同步镜像和异步镜像两种。同步镜像是指所有的对镜像磁盘或者逻辑卷的写操作,会同时发送命令和数据给镜像磁盘(或逻辑卷)对,且只有镜像磁盘对(或逻辑卷)的写操作都完成后才会通知写命令发出程序命令完成。异步镜像则是指所有的对镜像磁盘或者逻辑卷的写操作,当本地写操作执行完毕后,不必等镜像磁盘(或逻辑卷)的写操作完成,直接通知写命令发出程序命令完成。镜像磁盘(或逻辑卷)的写操作则通过后台进程(或线程)异步完成。远程镜像可以在应用程序和磁盘存储系统间进行命令和数据交互这条I/O路径的任何位置上实现,实现位置可以是:文件系统、卷管理器、设备驱动、主机适配卡(HBA)以及磁盘控制器。不同的位置有其各自的优缺点。

(3)快照技术　远程镜像技术往往同快照技术结合起来实现远程备份,比如通过镜像把数据备份到远程存储系统中,再用快照技术把远程存储系统中的信息备份到远程的**磁带库**、**光碟库**中。快照(snapshot)将时间引入了数据存储,应用服务器通过快照功能,在使用当前数据的同时,也可以看到所选择的以前某个时间点的数据。快照是对存储数据的"即时"(point-in-time)复制。它能够很方便地做到在线备份。在进行在线备份的时候,为了保证所有数据是同一时刻的,就需要对磁盘操作进行临时的冻结,并在冻结期间将数据复制并搬运到其他地方去。但由于数据量往往比较大,短时间内是无法完成如此多数据的复制和搬运的,这势必造成系统长时间无法使用。一个常用的办法是不实际复制数据,而是在此期间保留所有的数据更改历史,也就是在发生数据更新时,不直接覆盖原有数据,而是将存储管理中用来记录逻辑地址和物理地址对应关系的映射表进行复制,从而保留了快照时刻的一个逻辑

视图。快照技术可使用户在正常业务不受影响的情况下,实时提取当前在线业务数据。其“备份窗口”接近于零,可大大增加系统业务的连续性。

(4) 连续数据保护技术(continuous data protection,CDP)　周期性备份技术中,如果所采用的备份间隔时间为0或接近0,且需要保存多个备份状态的技术,则被称为连续数据保护技术。连续数据保护技术是主要用来实现零数据丢失的一种技术,它的原理是将任何数据更新都及时进行备份,从而在出现灾难时,能够恢复任何因灾难丢失的数据。

(5) 应用层容灾技术　主要包括专用的数据库容灾技术,如oracle data guard、IBM DB2 HADR、IBM Informix HDR、Quest SharePlex、DSG RealSync等,它们是通过对数据逻辑操作的复制,进行扫描和记录数据库日志实现。当然应用层容灾还包括一些是由各个应用程序在应用层各自单独实现的带语义的容灾方法。应用层容灾技术紧密与具体的应用程序相结合,对于特定应用的容灾问题都可以较好地解决,但主要缺点是与应用程序的语义太过相关,很难具有通用性。

(6) 全系统保护技术　近年来发展起来的全系统保护技术,逐渐呈现出了取代传统数据保护技术以及应用层容灾技术的趋势。所谓全系统备份,就是在进行容灾备份的时候,不仅复制和保存数据的状态,同时也复制和保存系统的完整运行状态,以便能够在进行恢复的时候全面恢复系统服务状态的一种容灾技术,全系统备份及恢复技术一般都需要操作系统级的虚拟化技术支持。

参考文献

1. 郑纬民. 数字化生存的基石和保障:数据存储与容灾技术. 中国科学技术前沿,2010, 12:177-226

2. Choy M. Leong H V. Wong M H. Disaster recovery techniques for database systems. Communications of ACM, 2000, 43(11):272-280

3. 李涛. 信息系统容灾抗毁原理与应用. 北京:人民邮电出版社,2007

(余宏亮)

ruancipan qudongqi

软磁盘驱动器(floppy disk drive, FDD)　磁盘驱动器的一种,驱动以圆形聚酯膜为基底、涂敷磁性层材料的可装卸软盘片,并实施数据存取的设备,简称软驱(参见磁盘存储器),如图1所示。世界上第一个5.25 in的软盘驱动器,是1976年由Shugart Associates公司为IBM的大型机研发的,后来用在IBM早期的个人计算机中。1980年,索尼公司推出了3.5 in的软盘驱动器。20世纪90年代初3.5 in、容量为1.44 MB的软盘一直是PC的标准配置。在很长一段时间里,计算机一般带有两个软驱,分别为5.25 in 1.2 M软驱和3.5 in 1.44 M软驱。目前,在个人计算机和笔记本电脑中软盘驱动器已经淘汰,其数据交换功能被移动U盘所取代。



图1　3.5 in软磁盘驱动器及软盘

工作原理　软盘驱动器的工作原理与基本结构类似于**硬磁盘驱动器**,由磁头、磁头驱动定位机构、软磁盘引导与夹紧装置、主轴、读写电路和控制电路等部分组成。这些部件用以完成软磁盘装卸、磁盘旋转、磁头寻找磁道、读写数据和状态检测等功能。

读写软磁盘数据信息工作过程　软盘通过引导机构插入后套在主轴驱动轮上,通过关门的连锁机构动作使压紧轮将软磁盘夹紧;加载机构使磁头与软磁盘接触;主轴电机带动主轴和软磁盘旋转,达到额定速度;由步进电机、丝杠、磁头小车和零道光电传感器组成的磁头驱动与定位机构将磁头移动到零道上。这时驱动器处于就绪状态,可以接受主机来的命令。

当软磁盘控制器接到主机命令后,经过对命令的解释,产生各种控制信号,第一步实现磁头寻道功能,使所选驱动器的磁头定位在目标磁道上。寻道前,磁头目前所在的磁道地址已经存放在道号寄存器中,目标磁道号也已经存于暂时寄存器中;比较当前磁道与目标磁道,求出磁头需要移动的磁道数和移动方向,由控制器给出驱动步进电机的步进脉冲

和方向信号,驱动磁头至目标磁道,依靠电机的电磁阻尼与传动机构的机械阻尼使磁头稳固地定位在目标磁道上。第二步检出索引,检测扇区地址,实现数据的读写功能。

3.5 in,双面、倍密度软磁盘的技术性能如表1所示。

表1 双面、倍密度软磁盘的技术性能

技术性能	数值
容量(格式化)/MB	1.44
数据传输率/(kb/s)	500
平均寻道时间/ms	60
转速/(r/min)	300
平均选转等待时间/ms	100
平均无故障间隔时间/h	10 000
外形尺寸	101 mm×150 mm×25 mm
重量/g	500

垂直记录软磁盘驱动器 此种驱动器在磁头、记录媒体、读写电路方面与纵向记录的软磁盘驱动器有较大的差别,其他部分基本类似。磁头与磁层的组合有三种方式:①环形头与钡铁氧化体涂敷媒体组合;②单磁头与 CoCr/Ni-Fe 非晶双层薄膜组合;③磁通闭合型单磁头与 CoCr/Ni-Fe 非晶双层薄膜组合。采用第一种方式的软盘驱动器用于微型计算机。它的未格式化容量为 4 MB(格式化容量为 2.88 MB);与普通的纵向记录软磁盘驱动器相比,其位密度高一倍。在其他参数相同时,它的数据传输率达 1 MB/s。此种驱动器因使用环形头对垂直取向媒体写入,读出是垂直分量较大,信号波形呈现不对称状态,需在读写电路中进行波形处理(例如使用 Hilbert 过滤器),或者在磁头前隙的后沿增设某种软磁薄膜。其他两种组合方式的软磁盘驱动器的记录密度虽然很高,但因媒体的耐磨性不佳,未出现成熟的产品。 (张江陵 周功业)

ruanjian ceshi

软件测试(software testing) 检测和评价软件以确定其质量的过程与方法,即评价软件或程序的属性和能力,以确定它是否满足所需结果的过程与方法。软件测试的通常意义是对软件进行动态评价,即用输入样例来运行程序,并比较实际输出和预期结果,但静态分析也是软件测试的组成部分。一次成功的测试是指揭示了迄今为止尚未发现的某些错误的测试,而不是指未能找出错误的测试。

软件测试可分为静态分析和动态测试。进行静态分析时,不必运行软件,只是通过对源代码进行分析,检测程序的控制流和数据流,以及发现执行不到的"死代码"、无限循环、未初始化的变量、未使用的数据、重复定义的数据等;也可能包括对多种复杂性度量值的计算。静态分析虽然不能取代动态测试,但它是动态测试开始前有用的质量检测手段。动态测试技术借助于输入样例来执行软件,一般又可分为功能测试(即黑盒测试)以及结构测试(即白盒测试)。

黑盒测试是基于软件功能的测试,它试图发现以下类型的错误:①功能不对或遗漏;②界面错误;③数据结构或外部数据库访问的错误;④性能错误;⑤初始化和终止错误。其测试用例设计依赖于软件的需求规约和**设计规约**,由于这种测试没有考虑程序内部代码结构,所以称为"黑盒"测试。

白盒测试在设计时考虑了程序内部代码结构,主要方法有:

(1) 基本路径测试 任何软件程序能表示成一个控制流图,程序中的如果语句或循环语句可以用带条件求值(判定)点的不同分支来表示。可以选择测试用例使得程序"从控制流图的开始到结束沿某一条执行路径"执行,下一个测试用例应选择控制流中前面至少有一条边没有走过的路径,这样的执行路径称为独立路径。程序中独立路径数目等于判定点数加1,因此,有可能做到覆盖所有独立路径的完全覆盖测试。

(2) 语句覆盖测试 语句覆盖率是通过测试用例集所执行到的语句数目除以程序中总的可执行语句数目。通常认为,应使所测试的模块达到100%的语句覆盖率,除非有合理的理由不能达到。然而,即使达到100%语句覆盖,在软件中还是会有许多未测试到的方面。

(3) 分支或判定覆盖测试 类似于语句覆盖,但它不是统计执行过的语句数目,而是统计执行过的分支或判定的数目。一个如果语句有真、假两个分支,情况语句和循环语句可等价为分支语句。100%的语句覆盖并不能保证100%的分支覆盖。

(4) 判定-条件覆盖测试 判定-条件覆盖通过执行一个判定中每个条件输出的真和假两者来达到。对于由三个比较条件组成的判定,为了达到判定-条件覆盖,用来确定整个判定的三个条件每

个都需要考虑真和假的情况,其实这里有两个测试用例便可做到(例如,使三者都为真和三者都为假)。然而,即使达到判定-条件覆盖,也未必达到判定覆盖。

(5)条件组合覆盖测试　不仅要求每个条件的求值为真和假,而且要求对所有可能的条件组合求值以达到判定条件组合覆盖。例如,在一个判定中有三个条件,则这三个条件的组合需要有8个测试用例。对于现代结构化编程语言,100%的条件组合覆盖可保证100%的分支和语句覆盖。

软件测试策略与**软件工程**过程相对应:

单元测试对应程序编码,主要采用白盒测试技术。

集成测试对应软件的设计,主要检测各个单元集成过程中相互之间的接口错误以及形成新的组合后功能中的错误,多用黑盒测试技术并辅以一些白盒测试技术。

确认测试对应系统需求,以确保软件符合所有功能、行为、性能的需求规约,确认测试只使用黑盒测试技术。

此外,在软件提交给用户方时,要进行验收测试。验收测试的目的是建立系统能正常工作的信任,而不是为了发现错误。其他级别的测试还有α测试和β测试,前者是在开发者场所软件实际使用的测试;后者是软件产品正式发布前由用户方在用户场所实际使用的测试。

参考文献

1. Marciniak J J. Encyclopedia of software engineering. Wiley, 1994

2. Pressman R S. Software engineering: a practitioner's approach. 4th ed. McGraw-Hill, 1997

(金茂忠)

ruanjian dingyi wangluo

软件定义网络(software defined network, SDN)　一种新型的网络架构实现模型,允许网络管理人员通过对底层网络功能的抽象接口来实施网络配置、编程和管理。其核心思想是将网络设备的控制面与数据面分离开来,并以开放编程接口的方式实现对网络功能的灵活控制,为网络管理及应用创新提供易编程的平台。典型的软件定义网络模型包含以下五个部分:

(1)数据平面　由网络转发设备组成,只负责根据设定的规则策略进行数据报文的处理和转发,不参与网络规则策略的制定。软件定义网络中数据平面功能简单,通过指定通信协议与控制平面通信,获取数据报文处理规则。传统的网络设备(**交换机**、**路由器**等)的固件是由设备制造商锁定和控制,SDN希望将网络控制与物理网络分离,从而摆脱固件对网络架构的限制,这样用户便可以像安装、升级软件一样对网络架构进行修改,满足用户对网络架构进行调整、扩容和升级的需求。与此同时,底层的交换机和路由器等硬件则无须替换,节省大量成本的同时,网络架构升级周期将大大缩短。

(2)控制平面　同传统的路由交换设备不同,软件定义网络的控制平面和数据平面不再紧耦合在一起。软件定义网络模型具有一个集中式的控制平面,作为独立的实体存在,通常以软件形式运行在控制器上。控制平面的主要工作是建立并维护数据平面网络设备上的规则策略。由于数据转发的功能已经被剥离到交换机上,控制器本身不受性能要求的限制,可以使用更加灵活的硬件平台搭建,从而提高效率、降低成本。

(3)网络应用层　在控制平面上进一步抽象网络功能,屏蔽底层网络细节,向上提供编程接口和网络抽象视图,由应用程序制定更高层次的规则策略。应用层针对其关心的网络资源进行灵活编程,对网络流量进行灵活操控。在应用层进行创新网络应用开发的实践研究包括网络安全隔离、虚拟网络、多路径路由、网络节能、网络可视化等。

(4)南向接口　控制平面和数据平面之间通过特定的网络协议进行通信,这类通信协议通常被称为“**南向接口**”。通过南向接口,控制平面向数据平面下发规则策略,数据平面向控制平面上报网络状态信息。通过制定标准化的南向接口,能够摆脱设备厂商对网络设备的锁定和控制。当前,最著名的南向接口是ONF倡导的OpenFlow协议。作为一个开放的协议,OpenFlow突破了传统网络设备厂商对设备能力接口的壁垒。OpenFlow解决了如何由控制平面把用于和数据流做匹配的表项下发给数据平面的转发设备的问题。

(5)北向接口　控制平面向上层应用提供的抽象编程接口通常称为“**北向接口**”。软件定义网络控制平面对底层网络资源进行全局抽象,网络应用通过控制器提供的开放接口进行编程,最终实现可编程网络以灵活操控网络流量。因此,控制平面的开放程度决定了为上层应用提供的网络资源的丰富性以及使用的灵活性。北向接口设计有赖于控制平

面和网络应用层的功能边界划分。控制平面除了与底层网络设备通信,还需跟踪基础网络资源(链路、端口、流量、CPU 等)状态,并对这些资源进行灵活抽象(如生成全局网络拓扑图),提供给应用层;同时还要把应用层下发的操控策略翻译成数据平面流表,更新给数据平面。

参考文献

徐立冰,腾云. 云计算和大数据时代网络技术揭秘. 北京:人民邮电出版社,2013 (李丹)

ruanjian duliang

软件度量(software measurement) 根据预定义的规则来定量描述软件实体(比如产品、过程或项目)属性的过程。

软件度量常被用来理解、预测、评估、控制和改善产品质量与开发过程,并进行项目管理。在进行软件度量之前,这些被定量描述的软件实体及其属性需要先被明确标识出来。对软件产品实体的质量而言,常见的属性有功能性(functionality)、可靠性(reliability)、可使用性(usability)、有效性(efficiency)、可维护性(maintainability)和可移植性(portability)等。这些质量属性是软件度量要定量描述的目标。

软件度量根据所基于的软件度量指标(software metrics)相关规则来进行测量,其测量结果是度量指标值(metric values)。软件度量指标分为直接度量指标(通常用于测量软件实体的内部属性)和间接度量指标(通常用于测量软件实体的外部属性)。直接度量指标是不依赖于对应于其他软件实体属性的任何度量指标,而间接度量指标则有依赖。比如软件项目实体的缺陷密度是一个间接度量指标,它是由这个项目的缺陷总数和这个项目的规模(比如总代码行数)的比率计算得来,而一个项目的缺陷总数或总代码行数则是直接度量指标。软件产品度量指标通常包括规模(比如总代码行数)、复杂性(比如 McCabe 圈复杂性)等。针对面向对象软件产品(比如一个类),度量指标有类耦合度、类内聚度等。

目标-问题-度量指标 GQM(goal-question-metric)范例通常被用来指导软件度量的过程。这个范例有六个步骤。①开发企业目标以及所关联的度量目标;②产生问题用来尽可能完整地去定量地定义这些目标;③标识度量指标用来回答这些问题并追踪过程和产品与目标的一致性;④开发数据收集的机制;⑤实时地收集、查验和分析数据以给项目提供用于修正的反馈信息;⑥后期分析数据以评价和目标的一致性以及对将来改进提供建议。

参考文献

1. Fenton N E, Pfleeger S L. Software metrics: a rigorous and practical approach. CRC Press, 2013

2. Lorenz M, Kidd J. Object-oriented software metrics. Prentice Hall, 1994

3. Van Solingen R, Berghout E. The goal/question/metric method. McGraw-Hill Education. 1999

(谢涛)

ruanjian fangfaxue

软件方法学(software methodology) 以软件方法为研究对象的学科。主要涉及指导软件设计的原理和原则,以及基于这些原理、原则的方法和技术。狭义的也指某种特定的软件设计指导原则和方法体系。不论何种含义,关注的中心问题是如何设计正确的软件和高效率地设计软件。

计算机硬件技术的进步,一方面为计算机在社会生活和人类活动各个方面的应用提供了物质基础,另一方面对软件也提出了越来越高的要求。软件日益增加的复杂程度促使软件方法学和软件工程的研究兴起,但是二者的侧重点不同。软件工程的研究侧重于借鉴传统工程学科,最终目的是把软件生产变成一门制造工程。而研究软件方法学的目的是寻求科学方法的指导,把软件开发活动置于坚实的基础上。但是在提出的初期,两者间的界限不是十分清楚。在以后的发展过程中,相互间的影响与渗透也处处可见。总体上讲,软件工程需要方法学的依据和指导;方法学依赖软件工程特别是环境工具来发挥实际效用。

其中程序设计方法学是最先发展起来的部分,它研究使程序设计提高质量和效率的方法和相关技术,特别着重于建立程序设计的方法学基础和各类程序设计方法。计算机科学的一些精粹分支是从此生长出来的,例如,有关形式语法和编译的严格基础、程序正确性证明、程序验证、抽象数据类型、形式语义理论、结构程序设计等。都是作为程序设计方法学的问题在 20 世纪 60 年代和 70 年代期间发展起来的。

软件方法学的分类和基本内容

从开发风范上看,有自顶向下的开发方法,自底向上的开发方法。在实际软件开发中,大都是两种

方法的结合,只不过是应用于开发的不同阶段和以何者为主而已。

从性质上看,有形式方法与非形式方法。形式方法是一种具有坚实数学基础的方法,从而允许对系统和开发过程作严格处理和论证,适用于那些对系统安全性要求极高的软件的开发。非形式方法则不把严格性作为其主要着眼点。

从适用范围来看,有整体性方法与局部性方法。适用于软件开发全过程的是整体性方法,自顶向下方法,自底向上方法,各种软件自动化方法等均为整体性方法。适用于开发过程具体阶段的为局部性方法,如需求分析阶段的各种需求分析方法,设计阶段的各种设计方法。

(1) 自顶向下方法　自顶向下是一种决策的策略。软件开发涉及决策问题:作什么决策、如何决策和决策顺序。

自顶向下方法在任何时刻所作的决定都是当时对整个设计影响最大的那些决定。如果把所有决定分组或者分级,那么决策顺序是首先作最高级的决定,然后依次地作较低级的决定。同级的决定则按照随机的顺序或者按别的方法。一个决定的级别是看它距离要达到的最终目的(软件的实际实现)的远近程度。从问题本身出发来看,或是由外(用户所见的)向内(系统的实现)看,以距离实现近的决定为低级决定,远的为高级决定。

在这个自顶向下的过程中,一个复杂的问题(任务)被分解成若干个较小较简单的问题(子任务)。并且一直继续下去,直到每个小问题(子任务)都简单到能够直接解决(实现)为止(参见**结构化程序设计**)。

(2) 自底向上方法　与自顶向下方法相反,首先作最低级的决定,其次较高级的决定,最后建立起整个系统。也就是首先对最基本的构件和系统的内部函数,然后逐步升级到有关外部函数的决策。在这个过程中,开发者手中有越来越复杂和强有力的构件可以用来构造更高级的构件,直到最后构成整个系统。

(3) 形式方法　形式方法的目的是把软件作为数学来重新发现。形式方法需要有一个健全的数学基础,典型地是由一个(建立在集合论、逻辑或是代数的基础上的)形式规约语言来给定。它提供了精确地定义以下概念的手段,如一致性、完全性、规约、实现、正确性等。它还提供手段以证明规约是可实现的,证明所建立的系统是正确地实现的,以及证明系统具有某些性质而无须通过实际运行来确定其行为。

形式方法被用来避免系统中的歧义性、不完全性、不一致性。在系统开发的早期使用形式方法有助于避免设计缺陷,否则这些缺陷会留到后来的测试、排错阶段才能发现,从而造成巨大耗费。在系统开发的后面阶段用形式方法,可以帮助确定系统实现的正确性及不同实现的等价性。

(4) 软件自动化方法　是指利用计算机使软件的设计实现自动化的方法和相关技术。

长久以来,人们就希望发展出一种技术,使得计算机的用户所需要具备的机器知识和程序设计知识尽可能地少,一项任务从提出到在计算机上最终获得解决所需人的参与尽可能地少。最理想的情况是发展出这样的软件自动化系统:它是面向最终用户的,即直接由最终用户使用;它是通用的,即对任何应用领域都适用;它是全自动的,即不需要人的干预和协助。

由于无法达到面向最终用户、通用、全自动的理想目标,一种办法是降低对一个目标的要求,以争取较好地实现其他两个目标。即只追求有限目标,使难度降低,于是有实现软件自动化的三种策略:①自底向上式策略,不追求直接面向最终用户,而是逐步向最终用户靠拢;②狭窄领域策略,不追求适用于广泛领域,而是更好地解决专门领域中的问题;③辅助式策略,不追求全自动,而是以设计者为主体,提供辅助工具。

软件自动化的实现途径。软件自动化系统从问题描述(即规约)出发,得到可执行程序,有如下四种实现途径(参见**软件自动化方法**):

过程途径,迄今最成功的途径。写一个专用的程序(编译程序或程序生成器),按部就班进行加工以得到结果。但因为系统的每个特色都要靠相应的专门算法来实现,此途径不适于实现面向广泛领域中的最终用户的软件自动化系统。

演绎途径,或演绎综合。要得到满足某个规约的程序,相当于寻求该规约可满足性的构造性证明,并根据此证明具体构造出一个算法来。这种方法具有健全的数学基础,因此很吸引人。任何演绎方法均可用来支持软件自动化,但是在实践上还没有一种能用来证明现实规模的复杂定理。原因在于:演绎本质上是搜索推理路径的问题,而这在本质上又是指数复杂度的。不能对付复杂问题是因为搜索方法的低效,因此,一个办法是让人来参与,但是这决

不比程序设计更容易。另一个严重问题是,它所寻求的构造性证明,往往对应的是低效算法。

转换途径,研究得最多的途径。对软件自动化系统的输入是甚高级语言(VHLL),即直接面向最终用户的非过程语言,施以一系列转换之后,得到低级的实现表示(可执行程序)。一个转换由三个部分组成:模式、条件、动作。找到符合的模式之后,校验条件,成立时施用动作。转换要求保持程序正确性不变。有两类转换:一类从高层次到低层次(垂直),将抽象变为具体;另外一类不改变抽象级别(水平),而是进行优化。转换一次次施行直到某个条件满足为止(例如找不到符合的模式)。在许多方面转换与证明步骤并无不同,因而所面临的问题也一样。有时希望人来干预,但人也未必就有很好的办法。有少部分转换非常之关键(决策性的),其余的则不是很重要。如能使人可以干预关键部分,而其余的则自动化,当可取得更佳效果。转换方法吸引人之处在于它提供了程序设计知识的一种极清楚的表达方法。

归纳途径,将归纳推理用于程序设计。将程序的行为的实例作为规约的部分描述,通过归纳推理得出关于其行为的规律作为程序的规约,并且从它综合出程序来。相当于对实例进行推广,得到一个类,以所给出的实例为其特殊情形,所得到的程序是针对整个类的。此途径的好处是对用户不要求程序设计知识,实例非常容易理解和修改。问题是有关的技术还不能解决现实规模复杂度的程序。一种退一步的做法是不企图自动地推广实例,而是提供一个开发环境来协助用户进行推广。

参考文献

1. Gries D. Programming methodology: a collection of articles by members of IFIP WG 2.3. New York: Springer-Verlag,1978

2. Wegner P. Research directions in software technology. Cambridge,MA: MIT Press,1980

3. Lowry M R, McCartney R D. Automating software design. Menlo Park: AAAI Press/The MIT Press,1991

(董榕美)

ruanjian fenxi

软件分析(software analysis) 一般是指系统化地使用信息对软件的某些潜在规则和性质进行估算的过程。

软件分析一般包括文档(包括图形表格文档和文字文档)分析(document analysis)和代码分析(code analysis)两类。文档分析是对应用需求文档、软件需求文档、设计文档,甚至是实现文档(包括注释)的分析,采用的方法通常包括静态和动态两种,静态方法涵盖形式技术评审(formal technical review)、走查(walkthrough)、检查(inspection)、审计(audit)、实体关系分析(entity-relation analysis)、因果关系分析(cause-and-effect analysis)等;动态方法包括符号执行(symbolic execution)、规约测试(specification testing)和仿真(simulation)等。代码分析在某种意义上就是程序分析(program analysis),主要通过分析源程序代码来预测某个程序可能的运行行为,或者通过分析代码的执行结果来判断源程序代码中可能存在的缺陷。这实际上就是静态分析(static analysis)和动态分析(dynamic analysis)的基本思想。静态分析是一种输入无关(input insensitive)活动,它主要通过分析源程序代码来确定程序在不同环境下执行时可能的行为,并判断这些行为是否是程序设计人员想要的;动态分析是一种输入有关(input sensitive)活动,它是结合程序的不同输入以及每种输入产生的行为来判断源程序代码中是否存在缺陷,以此来达到提升代码的品质。一般来说,静态分析是一种完备的分析,因为它考虑了所有可能的输入情况,而动态分析则是不完备的,因为它只考虑某个或某些特定的输入情况;但是动态分析更加精确,因为它每次检查的都是具体的实际的值,而静态分析由于采用了某种抽象(abstraction)或者由于程序中存在不可行路径(infeasible paths)等问题,导致精确性不高。一般说来,静态分析主要包括代码阅读(code reading)、控制流分析(control-flow analysis)、数据流分析(data-flow analysis)和类型分析(type analysis)等;动态分析主要包括路径剖析(path profiling)、执行轨迹跟踪(execution trace tracking)和软件测试(software testing)。

软件分析的目的是达到对软件的充分熟悉和理解,以便为改正软件缺陷、改进软件的功能和性能、提升软件的品质服务。软件分析的典型应用包括:①错误检测(error detection) 软件分析是检测错误的一种手段,比较适合发现语法语义错误和逻辑关系错误等;②程序纠错(program correct) 例如,针对因果关系模糊问题,就可以通过分析因果图来得到很好的解决;③程序优化(program optimization) 例如,通过重构程序的结构来提高程序的执行效率、提高可重用性等。

参考文献

1. 梅宏,王千祥,张路,王戟. 软件分析技术进展. 计算机学报,2009,32(9):1697-1710

2. Nielson F, Nielson H R, Hankin C. Principles of program analysis, Springer, 2004

3. Jackson D, Rinard M. Software analysis: a roadmap. In: Ghezzi C, Jazayeri M, Wolf A L (eds.) Proceedings of the 22nd International Conference on on Software Engineering, ICSE 2000, Limerick Ireland, June 4-11, 2000

(李必信)

ruanjian fuyong

软件复用(software reuse) 提高软件生产力和质量的一种技术,将已有软件的各种有关知识用于建立新的软件,以缩减软件开发和维护的花费。早期的软件复用主要是代码级复用,被复用的知识专指程序,后来扩大到包括领域知识、开发经验、设计决定、体系结构、需求、设计、代码和文档等一切有关方面。

发展简史

软件复用的最早例子是子程序库。但是通常认为,软件复用的正式提出是在 1968 年。D. McIlroy 在国际上首次讨论软件工程的会议上建议建立生产软组件的工厂,用产品目录上的软组件构成复杂系统,以作为解决“软件危机”的一种可能技术途径。

在 20 世纪 70 年代中开始了软件工厂的多项研究实验。80 年代软件复用技术得到美国、欧洲、日本的政府和企业界的倡导与支持,如美国先进研究计划署(ARPA)的 STARS 计划;欧洲信息技术研究战略计划支持的若干计划,特别是 Eureka 软件工厂;日本的若干软件工厂(Hitachi, Toshiba, NEC, Fujitsu 及 NTT 等);以及美国的若干公司级计划(GTE, AT&T, IBM, Hewlett-Packard)等。

80 年代后期在构件库系统、构件分类技术、可复用构件的创建与分发、复用支撑环境、公司级复用计划等方面取得了许多进展。但是没有达到原来对软件复用所预期的那种在软件生产力和质量上的重大提高,据认为是因为复用的范围太狭窄,许多对效率和质量关系重大的因素未在考虑之列。这导致了 80 年代末提出对软件复用的广义理解,使得与软件计划有关的一切,包括知识,均在复用之列。这就大大拓宽了研究领域,使复用问题无处不在。

最新的研究工作涉及非技术因素:管理、经济学、文化、法律。这些问题是重要的,是可能比技术问题更难以解决的。新的复用观把这些因素也包括在内。

软件复用技术的基本内容、现状和趋势

(1) 复用什么

思想、概念、算法:被复用的是对一类问题的一般性解决方法。以算法来说,寻求算法的研究已经相当成熟。但是从复用的角度看,还需要在发展和分发标准目录方面做更多工作,并且提供专门的有关如何复用算法的详细信息的基本目录。

制成品、构件:这是一直在着重做的。80 年代初以来,软件工厂和公司级的复用计划都在从事软构件复用。面向对象的技术在构件复用上非常受重视。构件复用研究工作着重点是质量和可靠性,构件的适应性也是重要问题。特定领域的构件集合是研究的新趋势。

过程、技能:最集中的研究领域之一,即如何把软件开发过程加以形式化并加以封装,创建可复用对象的集合。技能和技术诀窍的复用主要受到专家系统研究者的关注。

(2) 复用形式及范围

垂直式:在同一应用领域中的复用,目的是得出一族系统的通用模型,可以用作样板来装配新的系统。研究重点在领域分析和领域建模,使用面向对象的术语,就是在特定问题领域中标识出一类相似系统的对象及操作和建立领域的模型。大多数软件工厂和有长期计划的机构都集中力量于垂直式复用。

水平式:目的是在不同的应用中使用通用部件,科学子程序库、Unix 工具是这类复用的例子。主要问题在于部件的包装和表达,以及发展可供广泛接触的库和库的网络,为此需要建立部件分类标准和部件目录。

(3) 复用方式

有计划复用:系统化和正式的复用,软件工厂中的复用模式。有确定的指导原则和规程可以遵循,并且收集度量数据以评定复用的性能。这种复用需要真正来自高层的投资和承诺,而且难以预测投资的回报。还需要对现行的软件开发实践作重大的改变,对开发者有强制性的纪律要求,并且需要他们的妥协。核心问题是建立复用成熟性模型和经济学模型,以使经理人员了解预期效益、投资回报、初始代价和性能准则。

专有化复用:非正式的复用,通常也称为看机会的复用。这是现在发生得最多的情形,复用是个人行为,不是在课题一级。不存在规程,所使用的构

件也不是为复用目的而专门设计的。

(4) 实现复用的途径

合成途径：用已有构件作为新系统的积木块。基于完善地建立起来的积累、高效的库系统和标准接口。虽然标榜复用所有的软件产物，但是主要着重于源代码复用。主要问题是库系统（构件选择和检索）、领域分析（构件理解和构件适应）以及接口技术（构件集成）。

生成途径：在规约级的复用。通过应用代码生成器，可以提供具有最高潜力的回报。Unix 中语法分析器的生成器 Lex 和 YACC 是著名例子。主要问题是如何形式地表达针对特定领域的规约语言、软件进程和元生成器。

(5) 单元如何复用

黑箱：复用构件不经任何修改。可复用构件是封装了的，具有标准接口。这种途径保证了更高的质量和可靠性，但是创建这种可复用黑箱的代价比较昂贵。主要问题是验证和（发给证书）证明在任何可能情形下构件都完美无缺陷地执行。

白箱：构件可以修改使得适应于具体情形的需要。这是最常见的途径，特别是在专有化复用的场合。绝大多数复用计划，包括软件工厂都属于白箱复用。主要问题是构件的参数化和内部可适应性，以及把复用扩展到更高级产物如设计和规约。

(6) 何种工作产物被复用

源代码：现状是代码复用。已有的绝大多数复用工具、环境及方法是针对代码复用，但是发展趋势是越来越少强调代码复用。到最后，代码将从更高级的表达自动地生成。

设计：设计复用可提供比代码级复用更高的回报，但是仍是在生长中的技术。面向对象的方法可以部分地或间接地实现设计复用，但是系统化的实践仍在研究阶段。

规约：可以提供最高的回报，是生成式复用的焦点。当规约可供复用时，通常也就实现了设计级和代码级的复用。

对象：越来越受到重视的复用物。有多种方法、工具和环境来创建对象，面向对象的方法看来是未来的复用技术。趋势是走向集成化的工具和方法来覆盖开发全过程。

正文：即除代码之外的所有工作产物，特别是供人使用的文档。正文复用日益重要，趋势是把可复用正文与所有其他工作产物一起集成。

体系结构：最大的复用单位。要分析应用领域以找出共同的设计，然后用作基本样板来集成可复用部件或是开发专门的代码生成器。领域分析同时支持生成式和合成式复用。

参考文献

1. McIlroy D. Mass-produced software components. In: Buxton J M, et al. eds. Software Engineering Concepts and Techniques. 1968 NATO Conference of Software Engineering. New York: Petrocelli /Charter, 1969:88-98

2. Biggerstaff T J, Perlis A J. Software reusability, Volume Ⅰ: concepts and models, Volume Ⅱ: applications and experiences. New York: ACM Prees, 1989

3. Freeman P. Software reusability. New York: IEEE Computer Society Press, 1987

（董韫美）

ruanjian gongcheng

软件工程（software engineering） 应用计算机科学理论和技术以及工程管理原则和方法，按预算和进度实现满足用户要求的软件产品的工程，或以此为研究对象的学科。

发展简史

术语“软件工程”第一次出现在 1968 年的 NATO 会议上。这次会议以及而后的一系列会议集中讨论了大型软件开发项目中出现的诸如软件质量、开发成本以及按期交付等问题，以着手解决软件开发失控的所谓“软件危机”，并提出了一些克服“软件危机”的策略。

20 世纪 60 年代末至 80 年代初，围绕软件项目，开展了有关软件开发风范、开发方法以及支持工具的研究。其主要成果是，提出了**瀑布模型**、增量模型、**演化模型**、**螺旋模型**等，以期解决软件开发的逻辑问题；开发了一些**结构化程序设计语言**（例如 **PASCAL 语言**，**Ada 语言**），提出了结构化软件开发方法和面向数据结构的软件开发方法等，以期解决软件开发工具问题。随后，出现了一些管理方法和相应的支持工具。

随着软件系统规模的增大、复杂性的提高以及在关键领域应用的开展，20 世纪 80 年代以来，人们更加关注软件生产技术的研究和实践，并注重软件工程管理和软件质量的研究。其主要成果是，提出了软件生存周期过程概念，开发了一系列**能力成熟度模型**，开发了具有广泛应用的**面向对象语言**及相应的方法，开展了**计算机辅助软件工程**（CASE）的研究与实践，特别是开发了多种软件开发框架，并在

软件复用方面取得了有效成果。

基本内容

基于软件工程的定义,软件工程的框架可概括为三元组: 目标、活动和原则。

软件工程目标　软件工程目标是,生产具有功能性、可用性和开销合宜的产品。功能性是指软件产品达到预期功能需求的程度;可用性是指软件产品达到预期非功能需求的程度,包括性能、外部接口、质量属性以及设计约束等;开销合宜是指软件开发、运行的整个成本满足用户效能要求的程度。

软件工程活动　软件工程活动是生产一个最终达到工程目标的软件产品所需要的活动。基本活动(或称阶段)包括: 需求分析、设计、实现、验证与确认、维护。

(1) 需求分析是在一个抽象层上建立系统模型的活动。这一活动产生需求规约,确定了工程项目的范围,用以作为软件开发人员和客户之间契约的基础,并作为以后开发活动的输入。

(2) 设计定义了实现需求规约所需要的结构,包括软件体系结构以及详细的处理算法,即所谓**设计规约**,给出了实现软件需求的软件解决方案。

(3) 实现是由设计规约到代码的转换。为此,需要选择特定的语言和工具。

(4) 验证与确认是一项评估活动,其中主要包括需求规约、设计规约以及实现代码的评估。验证与确认可以是动态的或是静态的。测试是一种动态分析技术,以选定的输入来执行程序或程序段,并与预期结果进行比较;一般包括程序的单元测试、集成测试和系统测试等,就功能测试而言,采用的软件测试技术一是基于代码结构的白盒测试和基于规约的黑盒测试。静态评估是不执行程序的分析,以发现与预期目标的不一致性,例如软件评审、审计、代码“走查”以及程序的形式化验证等。

(5) 维护是在软件发布之后所进行的修改或开发,包括对发现错误的修正以及对环境的变化所进行的必要调整等。

软件工程原则　软件工程原则是为实现软件工程的三大目标,指导软件开发活动的行为准则。围绕软件开发,主要有以下四条基本原则:

(1) 选取适宜的开发风范　在系统开发中,为了适应需求的易变性,经常需要权衡软件需求、硬件需求以及其他因素之间的相互制约和影响,就要选用适宜的开发风范,涉及一定的开发准则和开发风格,以保证软件开发的逻辑性和可持续性,使最终的软件产品满足客户的要求。

(2) 采用合适的设计方法　在软件设计中,通常需要采用一种合适的软件设计方法,例如结构化方法和面向对象方法,或需要基于某一软件开发框架,以便体现诸如模块化、信息隐蔽、局部化、层次化、一致性、可跟踪性以及适应性等设计原则,使开发的软件达到软件工程的目标。

(3) 提供高质量的工程支持　软件工程如其他工程一样,需要提供高质量的工程支持,例如配置管理、质量保证等,才能按期交付高质量的软件产品。

(4) 有效的软件工程管理　软件工程的管理直接关系到生产产品机制-过程的定义,关系到可用资源的有效利用,关系到开发人员的技能培训以及相关利益方的参与等。因此,只有对软件过程实施有效管理,才能实现有效的软件工程。

这一软件工程框架确定了软件工程研究内容,主要包括: 软件生存周期过程,软件生存周期模型,**软件开发方法**,软件工程管理与支持技术,包括软件质量特征、过程及其改进以及计算机辅助软件工程(CASE)工具、环境等。

(1) 软件生存周期过程　软件生存周期中一系列相关的过程。每一个过程是一个特定的活动集合,而每一个活动是一个特定的任务集合,一个任务是将输入转化为输出的操作。

在这一方面的研究,其重要成果是著名的标准ISO/IEC 12207 软件生存周期过程。该标准自 1995 年发布以来历经多次演化,特别是近年来由于软件技术的快速发展以及软件开发在系统工程中扮演着越来越重要的角色,人们越来越从系统工程的视觉来审视软件生存周期过程,并采用良好的定义技术-框架来描述软件生存周期过程,形成了目前的 ISO/IEC 12207—2008 系统与软件生存周期过程,以便有效地支持系统工程中的软件过程定义、控制和改进。较详细的内容可参见**软件生存周期**过程。

(2) 软件生存周期模型　一个包括软件产品开发、运行和维护中有关过程、活动和任务的框架,用于指导软件工程的求解逻辑。

自提出软件危机以来,最先就开展这一方面的研究。到目前为止,其主要成果是提出了诸如**瀑布模型**、增量模型、**演化模型**、**螺旋模型**以及**喷泉模型**等。较详细的内容可参见**软件生存周期模型**。

尽管在软件工程实践中,基于软件生存周期过程和软件生存周期模型,每项工程都有自己特定的软件开发风范,但软件开发风范主要有四种,它们

是:瀑布风范,迭代风范,螺旋风范以及转换风范。

瀑布风范 一种基于**瀑布模型**的过程模式,该模式像“瀑布流水”那样来组织开发活动。每一活动有其确定的输入和输出——工作产品,可采用**结构化方法**或**面向对象方法**实施之。现已证明,该风范对需求不稳定的项目是不合适的,可能会出现大量“返工”想象,从而很难保障工程的合算性。

迭代风范 一种基于**演化模型**或增量模型而形成的一种有弹性的过程模式,该模式由一些小的开发步组成,每一步历经需求分析、设计、实现和验证,产生软件产品的一个增量。通过这些开发步的迭代,完成最终软件产品的开发。其中,一次迭代所产生的增量产品,往往发布给软件客户,以获取他们的反馈,用于指导相继的步骤。

螺旋风范 一种基于**螺旋模型**的过程模式,该模式关注问题求解逻辑,将**软件生存周期**的活动分为四个可重复的阶段:规划、风险分析、开发和评估。在早期螺旋循环中,经常使用原型构造,为最终的实现给出一些指导性决策。其中,评估和风险分析阶段都可针对项目是否继续作出一个决策;在规划阶段、风险分析阶段和开发阶段均进行需求规约活动,确定项目范围;设计和实现活动一般是在开发阶段进行; V&V 活动在开发阶段和评估阶段进行。

转换风范 一种基于待开发系统的形式化需求规约自动生成软件的过程模式,该模式通过一系列转换,将系统的形式化的需求规约转化为它的实现。

在以上研究成果的基础上,开展了一系列有关过程规划和过程改善方法的研究,以支持软件项目和开发者发现并运用适宜的过程。

(3) 软件开发方法 为了实施软件开发过程和活动,通常遵循的途径和指导。特别是系统分析和设计活动,其中的“途径”用于确定创建系统概念模型和解模型(解决方案)的特定抽象层次,而“指导”给出过程、活动或任务的基本行为原则、所采用的技术以及规程等。

在这一方面的研究,其典型成果有:**结构化方法**、**面向数据结构方法**、**面向对象方法**以及**维也纳开发方法**(VDM)等。

结构化方法 一种系统化的软件系统建模方法,包括结构化分析和结构化设计。结构化分析以分层的数据流图和控制流图为工具,开发系统的功能模型和数据模型。著名的有 E. Yourdon 和 T. DeMarco 的结构化分析(SSA)。结构化设计包括软件体系结构设计、过程(功能)设计和数据设计。体系结构设计以系统的功能模型为基础,逐层精化,最终形成系统的模块(子系统)以及它们之间的控制关系。其中,强调模块(子系统)的高内聚和低耦合。功能设计针对体系结构中的每一模块,给出它的过程属性描述,即算法设计。数据设计将系统的数据模型转化为数据结构。

面向数据结构方法 结构化方法的一种变体,其中,强调数据结构而不是数据流。这类方法的共同特征是:①标识关键的信息对象和操作;②以顺序、选择、重复三种基本结构层次式地表达数据结构;③提供一组步骤,把层次式的数据结构映射入程序结构。这类方法的范例有**数据结构化系统开发方法**(DSSD)或 Warnier/Orr 方法、**Jackson 系统开发**(JSD)方法。

面向对象方法 一种以对象、对象关系等来构造软件系统模型的系统化方法,主要涉及面向对象分析和设计。其中,对象由**封装**的属性和操作组成;对象类是具有相似属性、操作和关系的一组对象的描述。**类**是系统建模、系统设计以及实现等活动可复用的基础。这类方法著名的有:G Booch 的 OOD,J Rumbbaugh 的 OMT 以及 P. Coad 和 E. Yourdon 的 OOA,OOD 等。对象管理组织(OMG)发布的**统一建模语言**(UML)以及统一软件开发过程(USDP)受到了业界和学术界广泛关注,特别是 UML 以及 USDP 均有相应的工具,以便支持它们在软件开发中的应用。

(4) 软件工程管理与支持技术 按照有关质量体系的定义,即在制造和交付一个产品中,必须配以适当的技术作业程序和管理作业程序,这两种程序所形成的结构称为质量体系。可见,管理和支持技术是保障软件质量的不可或缺的成分。在软件工程管理与支持技术方面的研究,其主要成果有三:

一是发布了一个非常著名的国际标准:ISO/IEC 9126 软件质量模型。这一模型规约了软件产品和服务能够表明满足客户要求的一些性质和特征,例如功能正确性、方便性、易维护性、可测性、鲁棒性以及可用性等。为描述软件产品以及软件服务的需求提供了有效指导。

除了功能性需求之外,其他质量属性需求均是非功能需求。非功能需求还包括外部接口需求和设计约束等。非功能需求均作用于功能需求之上,即没有功能需求就没有什么非功能需求;并且一个非功能需求可对应多个功能需求。功能需求和非功能需求是需求规约的主要组成部分。

二是发布了一系列能力成熟度模型。最先发布的是软件工程能力成熟度模型，而后基于这一模型相继开发了个人能力成熟度模型、小组能力成熟度模型以及集成项目能力成熟度模型和系统工程能力成熟度模型等。为了简化评估工作，在国际标准化委员会主导下，把软件工程能力成熟度模型、集成项目能力成熟度模型和系统工程能力成熟度模型进行了科学整合，开发了目前经常使用的三个集成化能力成熟度模型，一个是针对开发的集成化能力成熟度模型（CMMI for development，2006，V1.2），一个是针对服务的集成化能力成熟度模型（CMMI for services，2007，V1.2），一个是针对获取的集成化能力成熟度模型（CMMI for acquisition，2007，V1.2）。人们常说的 CMMI 往往是指针对开发的集成化能力成熟度模型。

能力成熟度系列模型为开发组织软件开发过程的规划、实现并运行、评审和审计、维护和改进提供了一个框架，该框架将过程制度化程度分为五个等级：未执行级（初始级）、已管理级、已定义级、已定量管理级和持续优化级。有关能力成熟度模型系列标准的详细内容可参见有关条目。

一项软件工程管理一般涉及五大要素：人员、进度、质量、成本和范围（需求），管理的目的主要是围绕这五大要素进行“规划和组织”、“领导和控制”以及评估等，以便把以上五大要素平衡在一种可接受的状态。其中过程规划是软件工程管理的一项最基本任务，而过程改进是提高过程性能，包括提高软件产品质量的一种重要手段。另外，使用针对开发的集成化能力成熟度模型（CMMI for development）中的专用实践，可有效支持过程规划；使用其中的共用实践，可有效改善过程质量。

三是**计算机辅助软件工程**（CASE）工具与环境。自 1968 年软件工程概念提出以来，历经 40 多年的实践，人们发现，对于软件开发还需要更加灵活的，但又规范的工业化途径，包括贯穿软件产品整个开发过程的反馈和适当的调整。为此，有关政府和软件产业界共同努力，研究并开发了实现高质量大型软件系统的方法和工具。可用的 CASE 工具和环境的出现是使软件工程走向工业化途径的最好实践。

CASE 工具大体上可分为两类：一类是支持工程管理活动的工具，例如配置管理工具、成本估算工具、调度工具以及文档工具等；另一类是支持开发活动的工具，包括设计工具（如原型速成工具、建模工具等）、程序设计辅助工具（如调试工具、代码生成器等）、测试工具以及维护工具。

CASE 环境是 CASE 工具的集成，并协调地工作于同一环境中，支持整个软件开发工作。

展　望

在软件工程领域，未来仍将紧紧围绕提高软件质量和生产率问题，在工程技术方面和工程管理方面开展更加深入的研究。例如，①过程和软件质量特征测量技术，形式化分析技术以及测量工具；②软件可靠性技术；③逆向工程技术；④软件复用与集成技术等。随着以网络为中心体系结构的广泛应用，出现了一些新的计算风范（例如**云计算**）。在这样的计算环境中，就满足可用性这一工程目标，为软件工程技术和管理技术的研究和发展提出了一系列挑战，例如：软件系统及其信息的安全技术，可信计算技术等。

软件工程的研究迄今已经有了很大进展，为缓解“软件危机”发挥了重大作用。随着信息化需求，相信软件工程领域的研究将迎来一个崭新的时代。

参考文献

1. 杨芙清. 杨芙清文集. 北京：北京大学出版社，1998

2. Ralston A, et al. Encyclopedia of computer science. 4th ed. New York: Van Nostrand Reinhold, 2000

3. Marciniak J J. Encyclopedia of software Engineering. New York: John Wiley & Sons Inc. ,1994

（杨芙清）

ruanjian gongcheng jingjixue

软件工程经济学（software engineering economics） 从经济学的观点来研究、分析如何有效地开发、发布软件产品和支持用户使用这一产品的学科。

软件工程的目的是通过有关学科的应用使人们开发出来的软件系统成为对用户有用的产品。然而软件工程的效果是否良好不仅取决于计算机科学、软件方法学以及软件工具与环境的使用，也要看它是否满足经济学和社会效益方面的考虑。

软件工程经济学研究的问题包括：

（1）成本估算技术与成本估算模型的建立和使用　成本估算是**软件工程经济学**的中心问题。成本估算的经典方法是基于 B. W. Boehm 在 20 世纪 70 年代提出的结构化的成本模型（COCOMO）。使用

COCOMO 模型时的主要问题是需要对待开发的软件的大小(源程序行数)事先有所估计。进行这一估计需要其他方面的知识,例如依靠专家或拥有这类知识的专家系统,是 COCOMO 模型本身无法解决的。此外,由于近年来面向对象技术的广泛使用和软件部件可重用程度的提高,软件开发过程中新编写源程序的比重日益减少,而对日益庞大的部件库的内容进行搜索、了解所花工作量的比重日益增加,COCOMO 模型的这一缺点就更加明显,需要加以改进。

(2) 软件工程中不同决策的"成本-效益"分析　完成一个工程项目常有多种途径,项目负责人面对不同的方案需要对每种方案所需花费的成本和可能获得的效益进行分析和比较,以便选出最佳方案。

(3) 多目标决策分析　从经济学的观点对软件工程中要达到的多个(有时是互相冲突的)目标进行协调、作出决策以及对这些同时要达到的目标进行管理的技术。

(4) 不确定性的处理和风险分析　当我们利用某种模型或理论进行分析或预测时,不管这模型或理论多么完善,都需要有一些信息作为分析的出发点。然而在实际情况中,我们往往无法获得完整的信息,在这种情况下,分析的结果便带有不确定性,从而据此作出的决策便带有风险。软件工程经济学在这种场合中就要研究如下一类的问题:①分析比较按不同主观估计作出的决策之后所带来的风险的大小和性质;②研究是否值得为了获得更多的信息以降低风险而要多付出一些代价。

(5) 工期估计和控制　能否按时完成工程项目与缩短工期也是软件开发者经常面临的问题,因此工期估计和控制也是软件工程经济学要研究的重要课题。

软件产品或项目的功能规模是涉及软件开发和交易的成本、项目资源投入的预测、项目维护的预算、项目质量管理的要求以及产品上市的时间等方面的关键指标,因此,进行软件功能规模测量显得尤其重要。国际标准化组织(ISO)和国际电工委员会(IEC)联合技术委员会分别于 1998 年、2002 年和 2003 年推出了软件功能规模测量方面的系列标准,我国也注重这方面标准的研究与制订。国际标准化组织 ISO/IEC 相继发表了 4 个功能规模测量方法的标准,它们是:

—ISO/IEC 19761(COSMIC-FFP 方法,由公共软件测量国际联合会提出并维护);

—ISO/IEC 20926(IFPUG 方法,由国际功能点用户组提出并维护);

—ISO/IEC 20968(Mk Ⅱ方法,由英国软件度量协会提出并维护);

—ISO/IEC 24750(NESMA 方法,由荷兰软件度量用户协会提出并维护)。

其中,COSMIC-FFP 方法声明可适用于管理信息系统(MIS)和实时软件系统二类软件,IFPUG 方法声明可适于所有类型的软件,Mk Ⅱ方法声明可适用于逻辑事务能被确定的任何软件类型,NESMA 方法非常类似于 IFPUG 方法也可适用于所有软件类型。

参考文献

1. 巴里·W. 贝姆. 软件工程经济学. 赵越,等译. 北京:中国铁道出版社,1990

2. ISO/IEC 14143-6:2006 Information technology Software measurement—Functional size measurement—Part 6:Guide for use of ISO/IEC 14143 seriesand related standards. 2006　(周锡令　杨根兴)

ruanjian gongju

软件工具(software tool)　一类软件,用来辅助计算机软件的开发、运行、维护、管理、支持等过程中的活动或任务。使用软件工具能节省软件生产开销,提高软件生产率和产品质量。

早期人们构造软件,从本质上讲就是进行程序设计。引导程序、装入程序和编辑程序可看作是最早使用的软件工具。在**汇编语言**和高级**程序设计语言**出现以后,与之相应的汇编程序、解释程序、编译程序、连接程序和排错程序就成了当时用于开发软件的主要工具。以后,编辑程序从行编辑发展到全屏幕编辑,进而又产生了可以识别语言文法结构的语法制导结构化编辑程序。20 世纪 60 年代末出现了软件工程以后,支持软件需求分析、设计、编码、测试、维护和管理等活动的各种工具相继诞生,其中有的已能实现程序和文档的自动或半自动生成以及程序正确性的形式化验证。进入 20 世纪 80 年代后,随着图形用户界面(GUI)技术的发展,出现了用户界面工具(如窗口系统)。近年来,新颖的多媒体软件工具也应运而生。由于软件开发过程本身是由若干活动构成的,所以人们很自然地提出了把支持特定开发过程的单个工具集成在一起的要求。**工具集成**意味着把若干工具或工具构件结合起来,使其能够协同工作。20 世纪 80 年代中期,把一组软件工

具按一定的软件开发方法和过程模型有机地组织起来的集成化软件开发环境已开始出现。软件工具的种类繁多，可以从不同的观点来进行分类。由于大多数软件工具仅限于支持软件生存周期过程中的某些特定活动，所以通常把它们分成需求分析工具、概要设计工具、详细设计工具、编码工具、测试工具和维护理解工具等。软件管理和支持过程往往要跨越多个活动，支持这些活动的常用工具有：软件项目管理工具、软件配置管理工具、软件质量工具以及文档编辑工具、建模工具等。这些工具也常被称为计算机辅助软件工程(CASE)工具。

进入21世纪以来，伴随着通信与信息技术的快速发展，特别是互联网、移动通信的普及，推动了软件即服务(SaaS)等新型软件应用模式的形成，并陆续涌现出了网格计算(grid computing)、物联网(internet of things)、云计算(cloud computing)等计算模式和应用。全球化的文化交融、跨地域的协同、应需而变的敏捷开发、软件过程的持续改进、基于可复用构件的软件组装、软件生产线、开放式的软件服务支持等已成为软件开发、运行和维护的重要特征。因此，软件工具的用途和种类也在进一步拓展，支持软件国际化(如语言翻译、多语种的人机交互界面)、基于音视频和文字交流的协同工作平台、开源软件库和开源社区环境以及面向网格、物联网及云计算的基础软件和应用测试支撑工具等亦成为软件开发的重要支撑。软件工程环境(SEE)可以看作是一系列面向软件过程的服务，这些服务由相应的自动化或半自动化软件工具或工具构件来实现，用来支持在网络环境下从事需求工程、软件开发、再工程以及维护等各种软件工程活动。服务的范畴可分为软件工程服务(如软件建模)、技术管理服务(如重用、配置管理)、项目管理服务(如估算、项目监控)、过程管理服务(如过程监控、过程改进)、支持服务(如发布、策略执行)、基础设施服务(知识库、通信、操作系统服务)以及系统工程服务(如系统建模)和系统工程技术服务(如价值分析)。

以下列举几种典型的软件工具：

需求分析工具　支持需求分析活动。主要工具有：领域需求分析与建模工具、企事业过程建模工具、面向对象建模工具、数据流图(DFD图)工具、实体关系模型工具、状态转换图工具、原型工具、数据字典工具、模型检测工具以及分析和验证需求定义规约的工具等。

设计工具　支持各种设计活动。主要工具有：面向对象建模工具、软件体系结构设计与分析工具、程序结构图(SC图)设计工具、用户界面设计工具等。这些工具用于辅助建立软件的系统模型，包括其结构模型和行为模型，支持数据和控制流的模型化表示、数据内容(数据字典)定义、处理过程定义等，并支持设计模型的一致性等检查和验证，以防止把错误传播到设计或实现阶段。有些工具适用于软件开发的各个阶段，如面向对象建模工具等。

编程工具　支持编码活动。主要工具有文本编辑工具、语法制导结构化编辑工具、代码调试工具，编译程序、汇编程序、连接程序以及集成化的编程环境等。编程环境通常支持一种或多种编程语言(如Java、C ++ 、C、FORTRAN、Smalltalk、Eiffel 等)。常见的编程环境大多采用了图形化人机交互界面，不仅操作便捷，同时还集成了丰富的功能，如支持在大型构件库中检索和选择适用的构件等。此外，还有面向特定领域、应用或编程语言的软件开发与代码生成工具。这些工具提供了具有更高抽象层次的描述能力，它们能比常规程序设计语言在更高层次上抽象说明一个应用信息系统，不仅能把系统描述自动转换成程序，基于标准架构和可复用构件生成应用程序框架，而且能帮助校验系统规约的正确性，使其输出结果与用户需求相符合。

测试工具　支持各种测试活动。主要工具有：面向源程序代码的分析与测试工具，如静态分析工具、代码审查工具、动态覆盖率测试工具；面向软件各种外部特性的测试支持工具，如性能检测工具、鲁棒性测试工具、人机交互界面自动测试工具；面向不同应用类型的测试自动化支持工具，如模型驱动的分布式测试建模工具、测试用例和测试脚本生成工具、分布式测试执行支撑平台、嵌入式软件仿真测试工具、测试结果分析与评估工具、测试报告生成工具以及测试过程管理工具等。

静态分析工具通过对源程序的程序结构、数据流和控制流进行分析，可帮助分析和理解程序，发现一些隐藏的错误。在并行(并发)程序设计里，能辅助检测死锁和同步异常等情况。

代码审查工具采用一组针对编程规范和编程风格的检查规则对程序代码进行自动检查，可发现多达几十种甚至数百种违反编程规范和编程风格的代码缺陷，如变量未赋初值、函数参数个数或类型不匹配、指针越界、内存泄漏、命名不规范、缺少注释或注释不规范等。

覆盖率测试工具通过对程序的执行轨迹进行探

测，得到函数、语句、分支、条件、路径等程序基本单元的覆盖情况（执行次数），测试有关变量值的断言。

维护和理解工具 支持软件维护和理解活动。主要工具有：程序结构分析程序、文档分析工具、程序理解工具以及源程序至类图、控制流程图或数据流程图等的逆向分析与自动转换工具等。

逆向工程工具对已经开发完成的源程序进行分析，抽取程序的系统结构、控制结构、数据结构和数据流等信息，生成类图、程序结构图等。还有一类用来监控软件执行，并利用监控期间获取的信息来建立程序行为模型的动态逆向工程工具。

再工程工具用来支持重构一个功能和性能更为完善的改进的软件系统。如代码重构工具能重新构造程序代码，使之与程序设计要求相符；数据再工程工具能交互修改数据库的逻辑结构，并重构一个新的数据库物理设计。此外，还有一类构件分析与提取工具通过分析软件代码，辅助提炼可复用的软件构件，用以支持软件的重构和软件代码的复用。

项目管理工具 支持软件项目管理活动。这些工具把支持的重点放在一些必需关注的管理环节上。工具能对软件项目的工作量、成本和工期进行估算，可定义工作分解结构、工作调度计划以及对项目开发状况进行连续的跟踪和控制。此外，还能用工具对软件项目进行度量，提供软件生产率和软件产品质量的指标。

配置管理工具 支持软件的配置管理活动，能辅助完成软件配置项的标识、版本控制、变化控制、审计和状态统计等基本任务，使各配置项的存取、修改和系统生成易于实现，从而能简化审计过程，改进状态统计，减少错误，提高系统质量。

评价软件工具 支持软件评价活动，为提高软件的开发效率和质量提供辅助支持。

好的软件工具除了便于使用、工作可靠、性能良好和技术服务支持完善以外，还要考虑工具应能剪裁和定制，以适应环境变化以及特定用户的要求。此外，工具应易于安装到用户已有的环境和系统中，并与其他工具和数据库相匹配。

参考文献

1. Chikofsky E. Computer-aided software engineering (CASE). 2nd ed. Los Alamitos: IEEE Computer Society Press, 1993

2. Fisher A S. CASE-using software development tools. New York: John Wiley & Sons Inc., 1988

3. Gane C. Computer-aided software engineering: the methodologies, the products, and the future. Englewood Cliffs: Prentice Hall Inc., 1990

4. ISO/IEC TR 14471: 2007: Information technology—software engineering—Guidelines for the adoption of CASE tools. 2007

5. ISO/IEC 15940: 2006. Systems and Software Engineering—Software Engineering Environment Services. 2006

6. ISO/IEC 12207: 2008. Systems and software engineering—Software life cycle processes. 2008

（刘超）

ruanjian goujian

软件构件（software component） 软件系统中具有相对独立功能，可以明确辨识，接口由规约指定，与语境有明显依赖关系，可独立部署，且多由第三方提供的可组装软件实体。软件构件须承载有用的功能，并遵循某种构件模型。可复用构件是指具有可复用价值的构件。

当前，对软件构件的定义及其内涵尚未形成一致共识，不存在公认的构件模型。然而，一个基本的认识是：构件的接口规约中，除其功能规约外，还需包括其对外交互连接的描述和支持自省的自描述信息。现有的构件模型基本可分为3类：

（1）构件描述-分类模型 描述构件的所有对用户查找、理解、选择、适应性修改及使用构件有帮助的信息，以及所有对构件库管理者分类和管理构件及构件间关系有帮助的信息。其目的是使得构件易于在构件库中被有效、高效地分类、储存和检索，易于被用户理解和复用，这就是构件库的数据模型。代表性的工作有（可复用库互操作组织，RIG）提出的统一数据模型（UDM）和基本互操作性数据模型（BIDM）。

（2）构件规约-组装模型 描述构件的功能和行为规约（包括构件对外提供的功能和需要外界提供的功能）、构件应用的语境以及构件间的交互和组装。用于规约构件并在设计层次上组装构件。这类模型通常体现于**接口定义语言**（IDL）、构件描述语言（CDL）和软件体系结构描述语言（ADL）中，基本都遵循概念、内容和语境（3C）模型。

（3）构件实现模型 描述构件在源程序级或二进制目标码级的实现机制，用于指导人们以某种程序设计语言或以某种可执行单元的形式来实现

构件，也称基础设施模型。这类模型并不关心构件的内部实现方式，重点是构件的接口封装机制，不过，当前主流的构件实现模型均是基于面向对象技术扩展而来。典型的代表性工作有：微软公司的COM/DCOM，OMG 的 CORBA/CCM，SUN 公司的 JavaBeans 和 Enterprise JavaBeans。通常，构件实现模型是和支持构件运行的**软件中间件**平台紧密相关的。

由于大多数构件模型都是基于面向对象的概念，有时人们将构件简单地用作对象的同义词。然而，构件和对象是两个相对独立的概念。**对象**是封装了状态和行为并有独特标识的软件实体，通过**类**定义对象的行为和结构。而构件除了封装状态和行为外，至少还需提供其对外交互的连接点和供自省的接口，并可同时拥有多个接口。构件由构件类型或构件模板定义。在使用上，对象(类)可在语言层次上通过继承等白盒方式被使用，而构件的实现通常完全对外界隐蔽并且有时只是以二进制代码形式存在，只能采用黑盒使用方式。实际上，一个构件可以由一个或多个对象实现，甚至可以由非面向对象语言的传统方法来实现，只需对外提供符合构件实现模型的接口即可。

参考文献

1. Crnkovic I, Larsson M. Building reliable component-based systems. Artech House, July 2002

2. Szyperski C. Component software: beyond object-oriented programming. New York: Addison Wesley/ACM Press, 1998

3. Heineman G T, Councill W T. Component-based software engineering: putting the pieces together. Reading, MA: Addison-Wesley,2001 (梅宏　谢涛)

ruanjian goujianku

软件构件库(software component library) 存储和管理可复用软件构件及其有关描述信息，用以支持开发人员共享构件资源的**软件系统**。可复用软件构件包括软件源码、目标系统、软件服务、解决方案等不同类型的软件制品及其相关的、具有复用价值的软件需求规约、分析与设计文档、应用数据、测试计划、测试用例、用户指南等文档和信息资产。

软件构件库是实施**软件复用**的重要基础设施。软件构件库主要由构件库管理系统 CLMS 以及其中存储和管理的可复用软件构件、其他资产及其信息、数据构成。

构件库管理系统提供如下功能，以支撑软件资源共享和复用的要求：①构件实体的可靠存储；②构件描述、分类、检索；③用户管理、权限控制；④面向资产的版权管理、版本控制；⑤构件入库和出库复用过程管理；⑥构件质量反馈、评估和分析支持。

软件构件库作为软件复用和共享的核心机制，已有大量的研究和实践。一些国际组织也有相应的技术标准规范，如北大西洋公约组织(NATO)针对 NATO、NATO 参与国和项目承包商制定了一组软件复用标准，包括“可复用构件开发标准”、“可复用软件构件库管理标准”和“软件复用过程标准”；IEEE 组织为构件库互操作而建立的“基本互操作数据模型”BIDM (Basic Interoperability Data Model, IEEE Standard 1420.1,1995)；OMG 组织为可复用资产描述和打包建立的“可复用资产规范”RAS (Reusable Asset Specification)；规定对软件构件及其相关资源进行管理与控制所必需的管理信息模型的中国电子行业标准《软件构件管理 第 1 部分：管理信息模型》SJ/T11374—2007；在 IEEE 标准 1517“信息技术—软件生命周期过程—复用过程标准”中也对构件库及其管理应用过程有明确的规范。

伴随着基于网络的软件开发和开源软件的兴起，当前构件库系统的主要发展趋势包括：面向网络、自动获取可复用资源及其描述信息等，对互联网上的海量信息进行抽取、组织，丰富构件库的资源规模；强化构件资源质量评估和分析技术，建立可信的软件构件库；建立网络化、服务化的构件管理云平台并与开发环境紧密、智能集成。

参考文献

1. 杨芙清，梅宏，李克勤. 软件复用与软件构件技术. 电子学报,1999,27(2): 68-75

2. 谢冰，王亚沙，李戈，等. 面向复用的软件资产与过程管理. 北京：清华大学出版社,2008

(杨芙清　谢冰)

ruanjian guocheng

软件过程(software process) **软件生存周期**中的一系列相关过程。又称软件生存周期过程。过程是活动的集合，活动是任务的集合，任务要起到把输入加工成输出的作用。活动的执行可以是顺序的、重复的(迭代的)、并行的、嵌套的，或者是有条件地引发的。

细言之，软件过程有三层含义：一为个体含义，即指软件产品或系统在生存周期中的某一类活动的

集合,如软件开发过程,软件管理过程等;二为整体含义,即指软件产品或系统在所有上述含义下的软件过程的总体;三为工程含义,即指解决软件过程的工程,它应用软件工程的原则、方法来构造软件过程模型,并结合软件产品的具体要求进行实例化以及在用户环境的运作,以此进一步提高软件生产率,降低成本。

软件过程的工程含义体现在如下几个方面:

(1) 软件过程的概念已不仅仅局限于软件开发和维护工作,它已发展为系统的集成和软件产品的制作与生产。这是由于软件复用技术已经较为成熟,软件产品市场已具较大规模,对系统集成和软件产品供应有迫切需要。为此提出了获取过程和供应过程,从而反映了软件过程不仅要有工程视面,也要有合同视面(包括系统视面和用户视面)。

(2) 提高生产率和软件质量,其关键在于管理和支持能力。所以软件过程又特别重视管理活动和支持活动,提出了管理过程和支持过程,从而反映了软件过程中的管理视面。

(3) 由于区分了软件开发环境和业务运作环境,并且系统或软件产品也不一定全部自行开发,可从获取过程得到所需的软件,从而提出了运作过程,反映了软件过程中的运作视面。

(4) 软件过程研究的对象已扩展到从事软件活动的人的工作。不同的角色,其视面不同,所负责的软件过程亦不相同。如获取者或供应者,按他们的合同视面只负责获取过程或供应过程;管理者按其管理视面负责的是管理过程;用户和操作人员按运作视面负责的是运作过程;开发人员和维护人员按其工程视面负责的是开发过程和维护过程;介入支持过程的人员按他们支持的目标负责支持过程的某些工作。

软件过程中的各个过程和活动,是按照几个线条并行地完成的,不仅开发、运作和维护贯穿整个生存周期,而且管理、支持、获取、供应等过程也贯穿整个生存周期。

软件过程有各种分类方法。按性质分有基本过程、支持过程和组织过程;按特征分有管理过程、开发过程和综合过程;按照不同人员的工作内容来分有**管理过程**、**获取过程**、**供应过程**、**开发过程**、**运作过程**、**维护过程**、**支持过程**。不管哪种分类方法,把软件过程运用到具体机构和具体应用领域或具体项目时,会把各种过程和活动进行剪裁,从而形成本机构或具体项目的软件生存周期过程模型,这就是**剪裁过程**,另外,还可能涉及基础设施过程、改进过程和培训过程。

管理过程　软件生存周期中管理者所负责的一系列活动。负责对所从事的过程,例如获取、供应、开发和支持等过程的活动和任务进行管理;软件管理过程适用于必须对自己的过程进行管理的任何一方。

获取过程　获取者获得一个系统或软件产品的一系列活动和任务。它从确定获取该系统或软件产品的需求开始,经过招标准备、合同准备、谈判及修改、对供应方的监督等活动,直至验收完成方告结束。获得该系统或软件产品的机构可就部分或全部获取活动与某机构签订合同,后者根据获取过程开展相应的活动。二者皆可称作获取者。

供应过程　供应者为获取者提供软件产品的一系列活动。它从理解系统或软件产品的需求开始,经过准备投标、签订合同、制定计划、实施和控制、评审和评价等活动,直至交付完成。供应者是那些提供软件产品的机构。

开发过程　软件开发人员所负责的一系列活动,其目的是依据合同成功地开发并交付软件。当要开发新的系统,或对已有的系统进行版本升级以及对已有系统进行有开发活动的移植时,都要涉及开发过程。

运作过程　用户和操作人员在用户的业务运作环境中为了使系统或软件产品投入运行所进行的一系列有关的活动。此过程包括对系统或软件产品的运作活动和用户运作时对他的支持活动。此过程的目的是在软件开发过程完成后,将该系统从开发的环境转移到用户的业务运作环境中运作;在运用中对用户的要求提供帮助和咨询;并对运行效果作出评价。

维护过程　软件维护人员所负责的一系列活动,目的是在保持软件整体性能的同时修改它,使它达到某一需求,直到其退役才告终止。从维护方式上讲有三种维护:改正维护、适应维护以及改善维护。当软件由于错误、缺陷、问题需要改进和修改以及对相应文档进行修改时,都要涉及维护过程。

支持过程　有关各方按他们支持的目标负责的一系列相关活动。支持过程有助于系统或软件产品的质量的提高,有助于它们的顺利运作。这类软件可以被其他类软件过程或本类中的其他软件过程所使用。软件过程的各个活动均可使用支持过程。支持过程可由使用它们的机构来实施;或作为一种服

务,由一个独立的机构来实施;也可作为项目的一项规定内容,由客户来实施。一个支持过程中的活动,由实施该过程的机构负责。这类软件过程包括:文档过程、配置管理过程、质量保证过程、验证过程、确认过程、联合评审过程、审计过程、问题解决过程等。

剪裁过程　对软件过程和活动实施剪裁的过程。将一选定模型以及相关标准应用于某一领域或具体软件项目,形成该领域的模型及标准或该软件项目的各个软件过程和活动。最初选定的模型是相对抽象的,具有相对普遍性的,而所选标准是描述活动全集的,将它们针对某领域剪裁意味着形成该领域的相对特殊的模型及标准;将它们针对软件项目进行剪裁意味着形成该项目的软件过程和活动,这是剪裁过程的双重意义。

基础设施过程　建立和维护任何其他过程所需的基础设施的过程。基础设施可以包括硬件、软件、工具、技术、标准和开发、运行、维护的设施。软件过程中的许多过程都应当明确该过程的基础设施。定义并建立所需的基础设施,并在其他相关过程执行时维护所建立的基础设施,是本过程的主要活动。

改进过程　建立、评估、度量、控制和改进软件生存周期的过程。主要活动为:制定一套组织计划,评价相关过程并实施分析、改进活动。软件过程中的任何一个过程都可能涉及此过程。

培训过程　为系统或软件产品提供人员培训的过程。软件的获取、开发、操作或维护的效果主要取决于有关人员所具备的知识和熟练程度。因此,拟定人员培训计划并及早实施是非常必要的。本过程要完成的主要活动有:制定所需的人员计划及培训计划,开发培训资料以及实施培训活动等。

软件的普遍存在性意味着不能脱离系统去考虑软件及其设计过程,而应当作为系统和系统设计过程的组成部分以及能为大型系统的软件部分、单独的软件产品和软件服务提供一整套可理解的生存周期过程,国际标准化组织和国际电工委员于 2008 年发布了“ISO/IEC12207—2008 系统与软件工程软件生存周期过程”,其中最大的变化是更好地和“ISO/IEC15288—2008 系统与软件工程 系统生存周期过程”两个国际标准统一,以促进它们的结合使用。它将软件过程分成:合同过程、组织项目启用过程、项目过程、技术过程、软件实现过程、软件支持过程和软件重用过程。前四个是系统环境过程,后三个是软件特定过程。从系统视面和软件作为服务的视面来看,软件过程在完整性上得到了很大的改善。

参考文献

1. IEEE Standard for Developing Software Life Cycle Processes—IEEE Std. 1074, 1991

2. ISO/IEC 12207: 1995 Information Technology—Software—Part 1: Software Life-Cycle Process. 1995

3. システム开发取引の共通フレーム(SLCP-JCF94),1994

4. ISO/IEC 12207: 2008 Systems And Software Engineering: Software Life-Cycle Processes. 2008

(朱三元)

ruanjian guocheng moxing

软件过程模型(software process model)　对**软件过程**的抽象描述。它反映了软件过程的活动、制品以及资源的某些特征、属性以及过程间的关联,不同的软件过程模型关注的内容和角度不同,因此给过程建模和执行者提供的信息也不同。软件过程模型的描述方法可以是形式化的、半形式化的或者非形式化的。

软件过程模型的目的是建立一种表示方法或者语言,可以凝练和共享业内最好的过程实践和经验,在技术上可以分为宏观和微观两种流派。宏观的软件过程模型技术主要关注过程的体系结构、过程的外部行为以及过程如何满足高层次的组织体系和特征要求,同时这样的模型中通常会建立一些合适的度量,以对过程执行情况和过程产品质量进行合适的监控和管理;微观软件过程模型主要关注过程定义的精确、完整、详细和无歧义,过程缺陷的探测以及人机协同的支持,通常可以对过程的协同性、完备性、执行的符合性进行验证。宏观和微观的软件过程模型并不是相互排斥和孤立的,很多过程模型方法融合了两者,譬如用过程内部的结构和细节来验证和解释其表现的外部行为。

软件过程模型的发展历史几乎伴随了**软件工程**发展的历史,针对每一时期软件开发的特点,都有典型的、通用的软件过程模型出现,这些软件过程模型为各个时期的软件开发和工程活动提供一个通用的框架和指南,使得软件项目可以定义其合适的软件过程及其必要的细节,排序优先和关键的活动,分配合理的资源,并可以进行持续的过程改进。最典型和代表性的通用软件过程模型有:

(1) 瀑布模型(waterfall model)　1970 年由

Winston W. Royce 提出的,是最经典的软件过程模型,如图 1 所示。后来的许多软件过程模型都是在**瀑布模型**的基础上派生的。

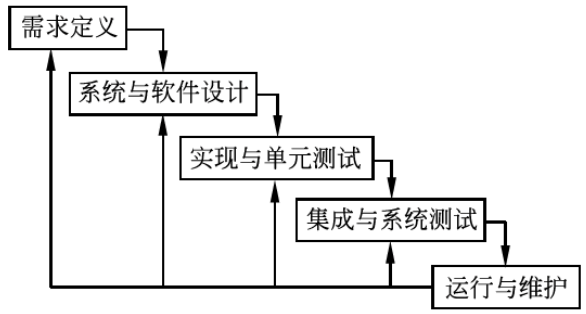


图 1　瀑布模型

瀑布模型的基本思想是软件生命周期每个阶段的产品都要被批准(approved),然后下一阶段才能开始,体现了质量管理理论中,过程不能非法转序的思想。但实践中,软件开发由于其特殊性,软件过程不可能是简单、严格的线性过程,其各个阶段必然有重叠和交互反馈,因此瀑布模型允许少量的迭代。但当阶段产品被批准后,该产品即被冻结,后续过程发现的问题,只能靠编程弥补,或者留待以后解决,甚至忽略这些问题。这样对阶段产品的正确性和用户满足性要求很高,但是由于用户需求、开发技术、工程师能力等因素的不确定性,这种质量性质的要求很难保证。特别是早期的阶段,如果冻结是不成熟的需求分析,就会导致最后交付的系统不是用户想要的;如果冻结的是不好的设计,就可能导致糟糕的系统结构,影响系统的后续演化和扩展。这种必须在过程的早期阶段的承诺,会影响到对客户需求变化的及时响应,这也是最初软件危机的根源。

(2)螺旋模型(spiral model)　1988 年由 Barry W. Boehm 在多年领导大型政府软件项目的实践中,应用并演化瀑布模型而提出的。如图 2 所示。

螺旋模型的基本概念是每一个螺旋周期都包含确定目标分方案、风险分析、验证、计划下一步这几

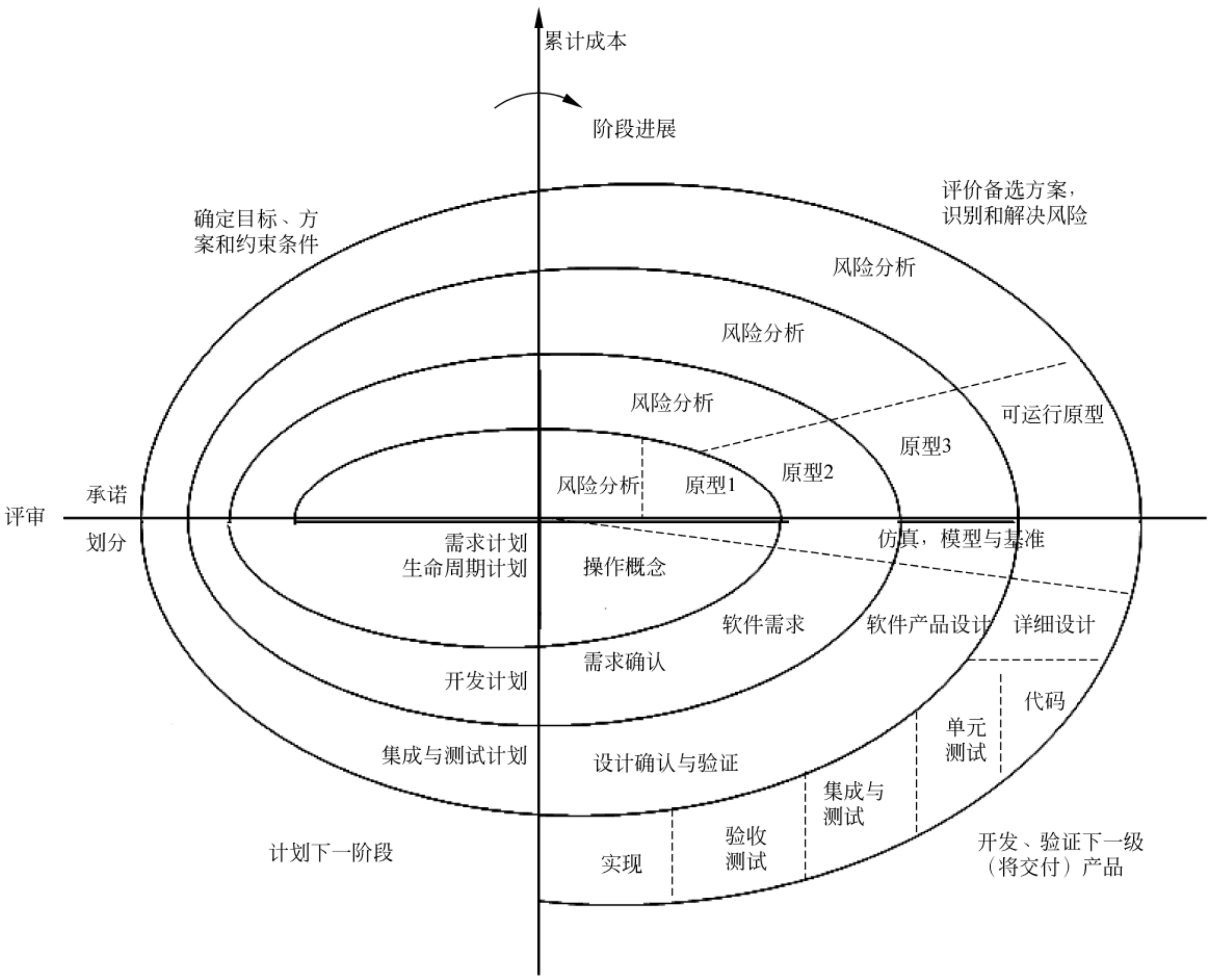


图 2　螺旋模型

类的活动,模型中的径向标注(radial dimension)表示到某个日期所完成活动的累计成本,角度标注(angular dimension)表示每一个螺旋周期的进展,亦即进行到哪个阶段。

每一个螺旋周期的起点是确定本周期的目标、实现方案和约束条件(如成本、进度、接口),然后进行风险分析和产品开发,每一个周期的结束必须要接受与产品相关的主要组织和人员的评审。螺旋模型的优势在于它吸纳了其他模型的优点,并以风险驱动的方式避免了诸如瀑布模型不能适应变化的缺点;其劣势在于非常依赖风险分析专家的经验。为克服早期模型的缺点,后期的螺旋模型发展为共赢螺旋模型(win-win spiral model)。

(3) 增量迭代模型(incremental iteration model) 有时也简称为迭代模型,其起源可以追溯到 20 世纪 50 年代,是早期瀑布模型改造或者替代方案,在许多大型项目中得到实践和应用,到 70 年代,瀑布模型由于其规范了软件开发的生命周期,被正式提出和受到人们的重视。而迭代模型到 21 世纪初期,由于极限编程思想的出现,被人们广泛重视,其应用范围也超过了瀑布模型,典型的变体就是 Rational 公司的 RUP 模型。迭代模型如图 3 所示。

迭代模型有许多优点,譬如客户不必等到整个系统出来,就可以逐步地了解系统,提出改进的需求,降低了整个项目失败的风险;并且可以优先交付高优先级的功能,快速占领用户市场。其缺点在于每次增量交付的规模比较小,难以合适地匹配客户需求和增量交付规模,而且一些基本的、公共的功能必须优先完成,但往往不能在迭代过程中很好地定义这些功能的细节要求。

(4) 基于构件的软件工程模型(component-based software engineering-CBSE Model) CBSE 模型基于已经存在大量可以重用的构件,软件系统的开发过程聚焦于如何集成这些构件,构成需要的软件系统。CBSE 模型如图 4 所示。图中,需求说明阶段和系统确认阶段类似于其他过程模型中对应的阶段,但中间阶段则是面向构件重用的过程,是其他过程模型中所没有的。

建立软件过程模型的目的是建立开发和演化过程中各种活动的秩序,并建立活动间过渡和转移的准则。最常见的建模方法有三种:工作流、数据流/活动流、角色/动作模型。

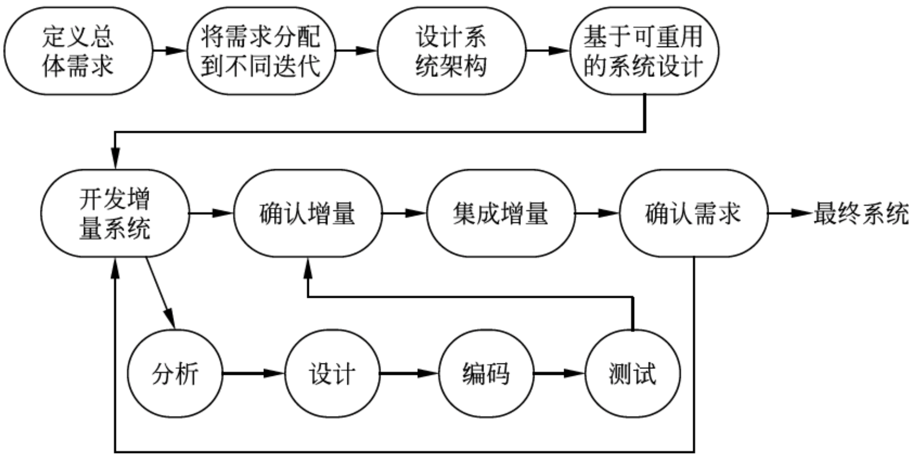


图 3　迭代模型

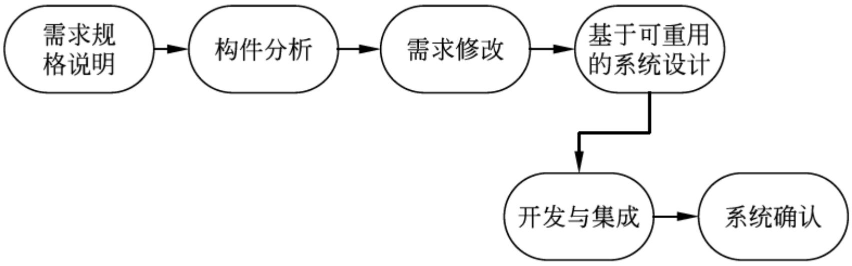


图 4　基于构件软件工程模型

工作流模型(workflow model) 主要描述过程中活动的顺序,活动的输入/输出以及活动间的依赖关系。工作流模型中活动代表人的动作,譬如软件需求分析、软件设计、软件实现和软件验证等。该类模型以活动间的输入/输出关系作为建模的主要约束关系。

数据流/活动流模型(dataflow or activity model) 将一组活动表示为一个过程,其中的每个活动都会实现一些信息的转化,模型主要描述如何将输入转化为输出,模型中的活动代表由人或者计算机实现的信息转化。该类模型主要以活动间的数据流转作为建模的主要约束关系。

角色/动作模型(role/action model) 主要描述人在软件过程各活动中的角色和责任,以角色关系作为建模的主要约束关系。

从 20 世纪 80 年代开始,软件过程技术进入快速发展的时期,以 CMM、SPICE 为代表的软件过程能力模型,为规模化软件生产和开发提供了有效的解决途径,代表了软件过程技术发展的趋势。其中,软件**能力成熟度模型**—CMM(capability maturity model)是卡内基梅隆大学软件工程研究院(Software Engineering Institute—SEI)为了满足美国联邦政府评估软件供应商能力的要求,于 1986 年开始研究的模型,并于 1991 年推出 CMM1.0 版,迄今已经发展到 CMMI(CMM-integrated)1.3 版,包含 CMMI-DEV(CMM for development),CMMI-SVC(CMMI for services),CMMI-ACQ(CMMI for acquisition)三个卫星模型,在世界范围内被广泛采纳和应用。CMMI 将软件过程组织为四类:项目管理过程、工程过程、支持过程和过程管理过程,提供了两种过程表示模型,即连续模型(continuous model)和分级模型(staged model),采用 CMMI 的组织可以选择其中一种表示模型来建立其过程模型。SPICE(software process improvement and capability dTermination)是由英国标准委员会(BSI,British Standards Institution)提出,ISO/IEC JTC1 于 1993 年正式建立的一个国际标准,标准号为 ISO/IEC 15504,从 2003 年逐步发布,目前包含 5 个部分。CMMI 和 SPICE 本质上是软件过程参考模型,他们提供了一组软件过程应该执行的实践和一种框架来组织这些实践,其中允许软件组织采取上面介绍的任何一种软件过程通用模型来组织和定义软件开发活动的生命周期。

软件过程实际上是人为中心的过程,为了配合实施 CMM,SEI 又提出了个人软件过程—PSP(personal software process)和团队软件过程—TSP(team software process)。PSP 关注个人,着重于对个体技能的改进和训练;TSP 关注小组和产品,着重于对小组性能的改进;CMM 则关注管理,着重于对组织能力的改进。

参考文献

1. 朱三元,钱乐秋,宿为民. 软件工程技术概论. 北京:科学出版社,2002

2. Sommerville L. Software engineering. 7th ed. Addison Wesley, 2004

3. Chrissis M B, Konrad M, Shrum S. CMMI for development: guideline for process integration and product improvement. 3rd ed. Addison Wesley, 2011

(王青)

ruanjian kaifa fangfa

软件开发方法(software development method) 软件开发过程所遵循的办法和步骤。软件开发活动的目的是有效地得到一些工作产物,也就是一个运行的系统及其支持文档,并且满足有关的质量要求。软件开发是一种非常复杂的脑力劳动,所以经常更多讨论的是软件开发方法学,指的是规则、方法和工具的集成,既支持开发,也支持以后的演化过程(交付运行后,系统还会变化:或是为了改错,或是为了功能的增减)。

关于组成软件开发和系统演化的活动有着各种模型(参见**软件生存周期**,**软件开发模型**,**软件过程**),但是典型地都包含了以下的过程或活动:分析、设计、实现、确认(测试验收)、演化(维护)。

有些软件开发方法是专门针对某一开发阶段的,属于局部性的软件开发方法。特别是软件开发的实践表明,在开发的早期阶段多做努力,在后来的测试和维护阶段就会使费用较大地得以缩减。因此,针对分析和设计阶段的软件开发方法特别受到重视。其他阶段的方法,从程序设计发展的初期起就是研究的重点,已经发展得比较成熟(参见**程序设计**,**维护过程**)。除了分阶段的局部性软件开发方法之外,还有覆盖开发全过程的全局性方法,尤为软件开发方法学注意的重点。

对软件开发方法的一般要求

当提出一种软件开发方法学时,应该考虑许多因素,包括:①覆盖开发全过程,并且便于在各阶段间的过渡;②便利于在开发各阶段中有关人之间的

通信；③支持有效的解决问题的技术；④支持系统设计和开发的各种不同途径；⑤在开发过程中支持软件正确性的校验和验证；⑥便于在系统需求中列入设计、实现和性能的约束；⑦支持设计师和其他技术人员的智力劳动；⑧在系统的整个生存周期都支持它的演化；⑨受自动化工具的支持。此外，在开发的所有阶段有关的软件产物都应该是可见和可控的；软件开发方法应该可教学，可转移；还应该是开放的，即可以容纳新的技术、管理方法和新工具，并且与已有的标准相适应。

当评价一种具体的软件开发方法时主要看四方面的特征。①技术特征：即支持各种技术概念的方法特色。如层次性、界面、控制流、数据抽象、过程抽象、并行性、安全性、正确性等。②使用特征：即用于具体开发情形时的有关特色。如可理解性、可转移性、可复用性、自动化工具的支持、生存周期的范围、任务范围、使用的广度、阶段过渡的易行性、对正确性的支持、可重复性、产品易修改性。③管理特征：即增强对软件开发活动管理的能力方面的特色。如可管理性、支持或是阻碍集体工作的程度、中间阶段的确定、工作产物、配置管理、阶段结束准则、计划性、费用估计等。④经济特征：即给软件开发机构产生的在质量和生产力方面的可见效益。如分阶段的局部效益、全生存周期效益、获得此方法的代价、使用它的代价、管理的代价等。

在一切方面都好的方法并不存在，也没有一种方法能适应于所有软件的开发需要。当需要选用一种软件开发方法时，可考虑以下的因素：①对该特定的软件开发方法是否已经具有经验，或者有受过训练的人员；②开发班子的组成情况；③为开发工作提供的环境如何；④任务计划管理的组织结构如何；⑤要解决的问题的领域性质；⑥开发工作时间进度框架；⑦是否有适当的自动化工具。

分析阶段使用的软件开发方法

需求分析阶段的任务是把软件的功用范围的一般性陈述精化为一个具体的规约，作为以后全部活动的根据。规约要表达出系统接受和产生什么数据、执行什么功能、确定了什么接口、施加了什么约束。也就是对问题及其解决的需求建立一个模型。模型主要刻画系统功能即“做什么”的问题，它在一定程度上还刻画软件结构，即由该软件的组成成分构成软件的方法和表示（参见**软件结构**）。

主要的方法有：

（1）结构化分析　使用得最广的需求建模方法，以数据流图和控制流图为基础，系统分析员划分出流变换函数，其次用状态迁移图来创建行为模型，用数据词典开发成数据模型。结构化分析最初是针对普通的数据处理应用发展起来的，起初是作为人工纸上作业的方法，以后发展为有自动化工具的支持。著名的有 E. Yourdon 和 T. DeMarco 的结构化系统分析（SSA），C. P. Gane 和 T. Sarson 的信息系统结构化分析设计及实现（STRADIS）。以后又发展出可用于实时系统开发的 P. T. Ward 和 S. J. Mellor 的软件工程需求分析（SERA）等方法。有许多软件工具支持结构化分析，以创建模型和帮助保证一致性和正确性（参见**结构化方法**）。

（2）面向数据结构方法　结构化方法的变种之一，着重于数据结构而不是数据流。这类方法的共同特征是：①协助系统分析员标识关键的信息对象和操作；②信息结构是层次式的；③数据结构的表达要求用顺序、选择、重复等合成构造；④提供一套步骤以把层次式数据结构映射入程序结构。这类方法的例子有数据结构化系统开发（DSSD）或 Warnier/Orr 方法，JSD 方法（参见 **Jackson 系统开发方法**）。

（3）面向对象分析（OOA）和数据建模　需求分析的面向对象方法是通过把类、对象、属性、操作作为最基本的构件来构造问题的模型。面向对象观点把对象的分类、属性继承、消息通信都组合在模型中。对象模型可以把问题的任何方面都标识为对象，特别是数据和进程。加工操作是对象的一部分，可通过向对象传送一个消息而启动。一旦定义了一个类，它就形成建模、设计、实现等各个级别的可复用性的基础，从一个类中可以实例化一个新对象。OOA 的主要目的就是标识出对象的类来（参见**面向对象方法**）。

数据建模可以看成是 OOA 的特例。它用实体-联系图作为主要的表达手段，数据建模着重在定义数据对象（不封装加工的操作），以及它们相互关联的方式。数据建模用于数据密集型的应用，也可以用作结构化分析的补充记法。

设计阶段使用的软件开发方法

设计阶段的任务是把需求翻译成软件的某种表示形式。它接受的是组成需求的信息模型、功能模型和行为模型，而产生数据设计、体系结构设计和过程设计作为工作结果。数据设计把信息模型转换成数据结构，体系结构设计定义程序的主要结构构件之间的关系，过程设计把结构构件转换为软件的过程描述。

从项目管理的观点，软件设计可以分为两步。初步设计专注于从需求到数据及软件体系结构的转换，详细设计专注于体系结构表达的求精，以便得到详细的数据结构和软件的算法表达。

主要的方法有结构化方法，面向数据结构的方法，面向对象的方法等。许多方法不仅仅用于需求分析阶段，而且也覆盖了以后的设计阶段。如属于结构化方法的结构化分析与设计技术（SADT），系统化活动建模方法（SAMM），W. P. Stevens，G. J. Myers 和 L. L. Constantine 的结构化设计（SD）等；属于面向数据结构方法的 JSD 方法，Warnier／Orr 方法（数据结构化系统开发 DSSD），程序的逻辑构造方法 LCP 等；面向对象的方法更是一种全局性方法，即覆盖了从分析、设计直到实现的开发方法。

形式化开发方法也是全局性方法中的重要一类，具有坚实的数学基础，被看成是开发正确的软件（安全性第一的系统）的有效方法。例如 VDM 方法和 Z 等（参见**形式方法**）。

尽可能地复用已有软件的各种有关知识于新软件开发活动的各个阶段，也是十分受重视的一种方法（参见**软件复用**）。

除此以外，许多新应用还涉及专门的用户界面设计活动，以建立人机交互机制。用户界面是进入交互式软件应用的门径，界面的面貌涉及多方面因素，因此用户界面设计是一个反复进行的过程。也就是建立一个设计模型，作为原型予以实现，交由用户校验，根据他们的意见进行修改。如此继续，直到满意为止。为此有许多界面设计和原型化工具，一般也称为用户界面工具集，或是用户界面开发系统。这些工具提供模块或对象，以便于创建诸如窗口、选单、设备交互、出错消息、命令，以及一个交互式环境的许多其他元素（参见**用户界面**、**用户界面管理系统**）。

参考文献

1. Birrell N D, Ould M A. A practical handbook for software development. Cambridge: Cambrigde University Press, 1985

2. Fisher A S. CASE using software development tools. 2nd ed. New York: John Wiley & Sons, 1991

（董韫美）

ruanjian kaifa huanjing

软件开发环境（software development environment） 支持软件产品开发的软件系统，简称 SDE。它由**软件工具**和环境集成机制构成，前者用以支持软件开发的相关过程、活动和任务，后者为工具集成和软件的开发、维护及管理提供统一的支持。

软件开发环境的发展，可以总结为五个发展阶段：①直接运行在裸机上的专用开发工具；②基于命令行的开发环境（command line environments，CLE）；③集成开发环境（integrated development environments，IDE）；④扩展开发环境（extended development environments，XDE）；⑤协同开发环境（collaborative development environments，CDE）。其中，前两阶段的工具提供各自独立的支持功能，比如编辑、编译、链接、调试、文档等功能。随后，工具箱（toolkit）的思想开始出现，工具箱将一组工具作为一个集合打包提供给开发人员，但是并不强调工具间的集成。

在 20 世纪 70 年代末期开始使用“环境”这一术语，其中最典型的是 Xerox 公司的面向对象语言 Smalltalk 程序设计环境，由此开始发展集成开发环境 IDE。常见的 IDE 是一个集成了编码、编译、调试、运行功能的开发环境。IDE 提供的特性主要聚焦于以语法为导向的编辑器扩展以及工具功能、数据的集成，用于增强用户的开发过程体验。

扩展开发环境 XDE 不仅支持时间上的松耦合开发，也支持空间上的分布开发，并且开始考虑对非软件编码直接依赖因素的管理。扩展开发环境提供的特性包括重构支持、需求变更管理、配置管理、建模等。XDE 的发展强调工具的集成和环境的扩展性。

协同开发环境 CDE 以协同开发思想为基础，更为强调多相关方、多参与者、多工具、多活动的协同开发支撑，通过整合协同机制与工具，从传统的面向软件开发人员的支撑机制扩展到了包括目标软件的行业领域专家、网络与硬件工程师等多相关方的协同开发，使得软件产品相关的所有利益相关方均能配置其感兴趣的属性，在互动的软件开发协作过程中及时沟通并做出对策，由此实现无缝的、完整的软件开发。当前 CDE 的协同特性包括虚拟团队、即时通信、网络会议、讨论等协同通信机制和包括配置管理、冲突检测、缺陷管理等的代码协同机制等方面，也需要进一步扩展对需求管理、项目管理、软件部署与运行监控等相关活动的协同支撑。

软件开发环境还可按以下几种角度分类：

按软件开发模型及开发方法分类，有支持**瀑布模型**、**演化模型**、**螺旋模型**、**喷泉模型**以及支持**结构化方法**、信息模型方法、**面向对象方法**等不同模型及

方法的软件开发环境。

按应用范围分类,有通用型和专用型软件开发环境。其中专用型软件开发环境与应用领域有关,故又可称为应用型软件开发环境。

按开发阶段分类,有前端开发环境(支持系统规划、分析、设计等阶段的活动)、后端开发环境(支持编程、测试等阶段的活动)、软件维护环境和逆向工程环境等。此类环境往往可通过对功能较全的环境进行剪裁而得到。

软件开发环境由软件工具集和集成机制两部分构成,软件工具和集成机制间的关系犹如"插件"和"插槽"间的关系。20 世纪 80 年代后期,NIST/ECMA 提出了集成化环境参考模型(见图 1)。

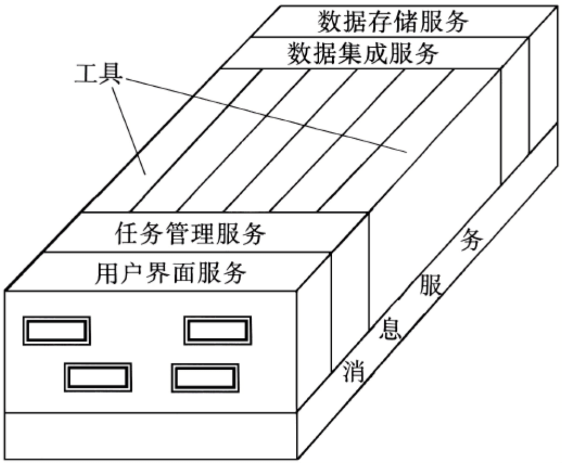


图 1　集成化环境参考模型

工具集　常用的工具可包括:支持特定过程模型和开发方法的工具,如支持瀑布模型及数据流方法的分析工具、设计工具、编码工具、测试工具、维护工具,支持面向对象方法的 OOA 工具、OOD 工具和 OOP 工具等;独立于模型和方法的相关工具,如界面辅助生成工具和文档出版工具;亦可包括项目管理、配置管理、文档管理、质量保证等管理类工具、针对特定领域的应用类工具以及支持协同的相关工具等。

集成机制　对工具的集成及用户软件的开发、维护及管理提供统一的支持。按功能可划分为环境信息库、过程控制及消息服务器、环境用户界面三个部分。

(1) 环境信息库　软件开发环境的核心,用以储存与系统开发有关的信息并支持信息的交流与共享,提供数据存储服务和数据集成服务。库中储存开发有关各类信息,一类是开发过程中产生的被开发系统的信息,如分析文档、设计文档、测试报告等;另一类是环境提供的支持信息,如文档模板、系统配置、过程模型、可复用构件等。常见的环境信息库包括配置版本库、文档库、缺陷跟踪库、测试数据库、通信信息库、可复用软件构件库等内容。

(2) 过程控制和消息服务器　实现过程集成及控制集成的基础。过程集成是按照具体软件开发过程的要求进行工具的选择与组合,控制集成实行工具之间的通信和协作工作。

(3) 环境用户界面　包括环境总界面和由它实行统一控制的各环境部件及工具的界面。统一的、具有一致视感(look & feel)的用户界面是软件开发环境的重要特征,是充分发挥环境的优越性、高效地使用工具并减轻用户的学习负担的保证。

当前,对软件开发环境的研究与实践仍在不断深入,新一代的软件开发环境将在以下几个方面取得进展:

(1) 开发工具和环境功能的智能化。

(2) 软件开发过程的可视化管理和定量分析优化支持。

(3) 服务化工具、基于服务组装的工具集成机制,以及由此构成的网络化开发环境。

参考文献

1. Chikofsky E. Computer-aided software engineering (CASE). 2nd ed. Los Alamitos: IEEE Computer Society Press, 1993

2. Norman R, Minder C. Integrated CASE. Theme issue of IEEE Software. Los Alamitos: IEEE Computer Society, 1992

3. Booch G, Brown A. Collaborative development environments. Advances in Computers, vol. 59. Academic Press, 2003

4. 杨芙清,邵维忠,梅宏. 面向对象的 CASE 环境青鸟Ⅱ型系统的设计与实现. 中国科学(A 辑), 1995,25(5):533-542　　(杨芙清　谢冰)

ruanjian kaifa moxing

软件开发模型(software development model)　软件开发全部过程、活动和任务的结构框架。软件开发包括需求、设计、编码和测试等阶段,有时也包括维护阶段。

软件开发模型能清晰、直观地表达软件开发全过程,明确规定了要完成的主要活动和任务,用来作为软件项目工作的基础。对于不同的应用系统、采

用不同的开发手段和方法,使用各种不同的程序设计语言以及各种不同技能的人员参与工作,它还应允许采用不同的软件工具或各种不同的软件工程环境。模型都应该是稳定有效和普遍适用的。

最早出现的软件开发模型是 1970 年 W. Royce 提出的瀑布模型。该模型给出了固定的顺序,将生存期活动从上一阶段向下一阶段逐级过渡,如同流水下泻,最终得到所开发的软件产品,投入使用。但实践表明,各个阶段间的关系并非如此简单。由于阶段评审可能出现向前阶段的反馈,致使在各阶段间产生环路,瀑布流水出现上流。瀑布模型为软件开发与维护提供了一种有效的管理模式,根据这一模式制订开发计划、进行成本预算、组织开发人员,以阶段评审和文档控制为手段有效地对整个开发过程进行指导,从而保证了软件产品的质量。瀑布模型 20 多年来之所以广为流行,是因为它在支持开发结构化软件、控制软件的开发复杂度、促进软件开发工程化方面起着显著作用。与此同时,瀑布模型在大量软件开发实践中也逐渐暴露出它的缺点。其中最为突出的缺点是该模型缺乏灵活性,无法通过开发活动澄清本来不够确切的软件需求。这些问题可能导致开发出的软件并不是用户真正需要的软件,并且这一点在开发过程完成后才有所察觉。面对这些情况,无疑要进行返工或是不得不在维护中纠正需求的偏差。但无论上述哪种情况都必须付出高额的代价,并将为软件开发带来不必要的损失。另一方面,随着软件开发项目规模的日益庞大,由于该模型不够灵活等缺点引发出的上述问题显得更为严重。

为弥补瀑布模型的不足,近年来已经提出了多种其他模型。常见的有:

演化模型 软件开发实践表明,许多开发项目由于人们对软件需求的认识模糊,很难一次开发成功。因而,返工再开发难以避免,常常要作两次开发,其产品才能令用户满意。第一次作试验开发,其目标只是在于探索可行性,弄清需求,第二次则在此基础上获得较为满意的产品。通常称第一次试验性产品为“原型”。演化模型在克服瀑布模型缺点、减少由于软件需求不明确给开发工作带来风险方面,确有显著效果。软件系统的原型有多种形式:

(1) 丢弃型——原型开发后,已获取了更为清晰的需求信息,原型无须保留而废弃;

(2) 演示型——开发原型仅以演示为目标;

(3) 样品型——原型规模与最终产品相同,只是原型仅供研究用;

(4) 增长式演化型——原型作为软件最终产品的一部分,可满足用户的部分需求,进一步在此基础上开发,则可增加需求,实现后再次交付使用;

(5) 粗陋型——用较短时间开发的简易原型。

螺旋模型 将瀑布模型与演化模型相结合,并且增加了两者所忽略的风险分析。螺旋模型通常用以指导大型软件项目的开发,它将软件项目开发分别划分为制订计划、风险分析、实施开发以及客户评估四类活动。沿着螺线每旋转一圈,表示开发出一个更为完善的新软件版本。如果开发风险过大,开发者和客户无法承受,项目有可能因此终止。多数情况下会沿着螺线继续下去,自内向外逐步延伸,最终得到满意的软件。

喷泉模型 喷泉一词本身体现了迭代和无间隙特性。系统某个部分常常重复工作多次,相关功能在每次迭代中随之加入演进的系统。所谓无间隙指在开发活动,即分析、设计和编码之间无明显边界。

智能模型 也称为基于知识的软件开发模型,它综合了上述若干模型,并得到专家系统的支持。该模型应用基于规则的系统,采用归约和推理机制,帮助软件人员完成开发工作,并使维护在系统规约一级进行。为此,开辟了知识库,将模型本身、软件工程知识与特定领域的知识分别存入数据库。以软件工程知识为基础的生成规则构成的专家系统与含有应用领域知识规则的其他专家系统相结合,构成了这一应用领域软件的开发系统。

参考文献

1. Marciniak J J. Encyclopedia of software engineering. John Wiley & Sons Inc. ,1994

2. McDermid J. Software engineer's reference book. Butterworth-Heinemann Ltd. ,1992 (郑人杰)

ruanjian lijie

软件理解(software understanding; software comprehension) 通过数据收集和分析来了解(特别是在设计文档不完全的情况下)软件的功能及其实现机理的方法和过程。

软件理解是软件维护中的重要工作。软件投入使用后,往往需要不断演进,以修正错误、扩充功能和适应新环境。对现有软件的理解是进行这一工作的前提。

软件理解一般有四项任务:识别程序单位、跟踪控制流、跟踪数据流以及综合程序逻辑。具体可

以分为四级：

（1）实现级　分析检查单个的程序结构，程序典型地表示成抽象语法树（AST）、符号表或普通源正文。

（2）结构级　分析检查程序结构过程中的结构关系，明确表示程序组成部分之间的依赖关系。

（3）功能级　分析检查程序结构和行为（功能）之间的关系，同时研究完成程序构作的合理性。

（4）应用级　检查特定于应用领域的概念。

软件理解是一项复杂任务，需要借助工具来进行，可用的工具种类很多，参见**理解工具**。

参考文献

李必信，郑国梁，李宣东，等. 软件理解研究与进展. 计算机研究与发展，1999，36(8)　　（郑国梁）

ruanjian liushui

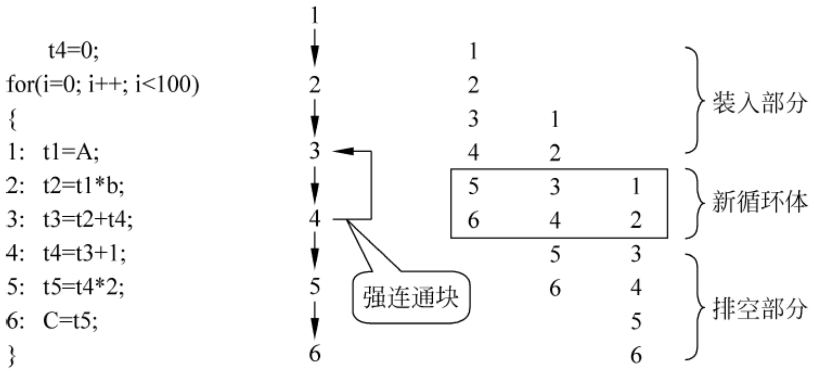
软件流水（software pipelining）　一种开发循环程序指令级并行性的方法。软件流水的基本思想是：在资源限制、数据相关、控制相关和周期性条件的保证下，一个循环可以等价变形，使循环的一次执行可以在前一次执行结束之前启动，从而使多个循环体按照流水线方式并行执行，因此称为软件流水。

在一般程序中，循环占据了绝大部分的处理机执行时间，因此，缩短循环程序的执行时间非常重要。软件流水能够消除循环程序中的绝大多数数据相关，大幅度提高循环程序的指令级并行性，充分利用硬件资源，加速循环程序的执行。

美国的 J. A. Fisher 于 1981 年提出了第一个软件流水算法，称为路径调度法。从那之后的 20 多年来，世界上出现了很多种软件流水算法，归纳起来，可以分为两大类：**模调度法**和**核心识别法**。在模调度法中，首先根据循环体内的数据相关和资源限制等条件确定循环体的启动间距，然后按照这个固定的间距顺序展开多个循环体，最后收拢被展开的循环体，形成高并行度的新循环体，同时生成装入部分和排空部分。核心识别法需要重复展开并调度多个循环体，直到发现重复模式为止，这个重复模式部分就是新循环体。Fisher 等人提出的路径调度法属于核心识别法。

图 1(a) 是一个简单的 C 语言程序，循环体内共有 6 个操作。从图 1(b) 中可以看出，由于所有相邻的操作之间都存在有写读数据相关，所以 6 个操作只能串行执行。假设每个操作的执行时间均为 1 个时钟周期，则执行一个循环体需要 6 个时钟周期。图 1(c) 是采用模调度法的软件流水结果。假设处理机内有加法器、乘法器和访问存储器部件各一个，它们能够独立并行工作，延迟时间均为 1 个时钟周期。由于在操作 3 与操作 4 之间存在一个强连通块，即同一个循环体内的操作 3 与操作 4 之间存在有关于 t3 的写读数据相关，相邻两个循环体的操作 4 与操作 3 之间存在有关于 t4 的写读数据相关，从而形成一个封闭的写读相关回路，因此，启动间距为 2。从图 1(c) 中看出，开头 4 个时钟周期为软件流水的装入部分，从第 5 个时钟周期开始，直到第 196 个时钟周期为止是新循环体，这时，加法器、乘法器和访问存储器部件同时工作，每 2 个时钟周期就能够执行完成一个循环体。最后附加 4 个时钟周期的排空部分。

对于基本块（循环次数确定，单入口单出口，循环体内没有条件分支、调用和中断等操作），目前的许多软件流水方法都能够得到最优或接近最优的结果。当循环体内包含有条件分支操作时，称为全局软件流水。与基本块软件流水方法相比，全局软件流水方法要复杂得多。到目前为止，已经出现了数



(a) 程序例子　(b) 数据相关图　(c) 软件流水（启动间距II=2）

图 1　一个软件流水例子

十种全局软件流水算法,归纳起来有如下三大类,分别是条件执行、条件组合和条件预测。对于条件执行方法,首先把条件分支上的操作转换成包含有条件的操作,即把控制相关转换成数据相关,然后采用基本块软件流水方法进行调度,最后通过硬件的支持来执行那些带有条件的操作。条件执行方法最早出现于 Cydra-5 处理机中,目前已经在著名的通用处理机安腾(Itanium)和著名的嵌入式处理机 ARM 等中得到应用。条件组合要求在软件流水过程中对条件分支上的操作按照启动间距重新进行折叠和组合,并且生成新的条件转移路径。条件组合方法的优点是不需要硬件支持,其缺点是代码的膨胀很大,往往是指数级的。条件预测方法是选择转移概率比较高的一条路径作为主路径进行软件流水,它能够保证主路径上的指令级并行度,牺牲其他路径上的指令级并行度。

软件流水通常需要有适当的硬件支持,除了需要有多个操作部件、多个指令译码器之外,还需要对循环控制、条件分支、寄存器换名和装入排空等给予支持。

总之,采用软件流水可以消除循环程序中的绝大部分数据相关,只需要保留占很小比例的强连通块数据相关,因此,其指令级并行度很高。与循环展开方法相比,软件流水方法的代码膨胀很小。

参考文献

Fisher J A, Rau B R. Instruct-level parallel processing. Science, 1991, 253: 1233-1241 (汤志忠)

ruanjian nixiang gongcheng

软件逆向工程(software reverse engineering) 分析软件系统,确定其构成成分及各成分间的关系,提取并生成系统抽象和设计信息的工程。

软件逆向工程包含数据收集、知识组织和信息浏览三项规范活动。这三者构成软件逆向工程的基本过程。

数据收集 本活动中所收集的数据是指作为学习、推理和讨论基础的实际信息。原始数据是构作和浏览高层抽象的基础,因而数据收集是软件逆向工程的一项基本活动。

知识组织 本活动中的知识是指所知内容的总和,包括数据以及从数据中导出的关系和规则。所收集的数据必须按某种数据模型保存起来,以便实现有效的存储和检索,从而帮助分析人员实现对对象及其关系的分析。

信息浏览 遍历表征目标系统信息的多维空间,按领域相关的标准分析和过滤信息,并以多种机制表达所得的信息,从而帮助分析人员进行程序理解过程中"假设—验证"的迭代过程。

软件逆向工程对软件测试、维护、鉴别、理解等诸多方面都有重要的辅助作用。(参见**软件再工程**)。

参考文献

1. Chikofsky E J, Cross J H. Reverse engineering and design recovery: a taxonomy. IEEE Software, 1990,7(1):13-17

2. Tilley S R, Paul S, Smith D B. Towards a framework for program understanding. In: Proceedings of the 4th Workshop on Program Comprehension (WPC' 96). IEEE Computer Society Press, 1996:19-28

3. Tilley S R. Coming attractions in program understanding II: highlights of 1997 and opportunities in 1998. Technical Report, CMU/SEI-98-TR-001, Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, February 1998 (孙家骕)

ruanjian peizhi guanli

软件配置管理(software configuration management) 一种按规则实施的管理软件**开发过程**和**维护过程**及其中间产品的方法。软件配置管理是**软件生存周期中支持过程**的一个关键组成部分,主要管理软件资源(包括与软件产品有关的各类文档、代码、数据、测试案例)及其演化,即管理软件生存周期中软件系统的构造和演化问题。

软件配置管理在整个软件生存周期中标识、定义系统中的软件配置项,系统化地控制对配置项的更改,记录和报告配置项的状态和修改申请,保证配置项的完整性、一致性和正确性,以及控制配置项的储存、装载和提交。其中,软件配置是指一个软件产品在软件生存周期各个阶段所产生的、机器可读或人工可读的各种形式和各种版本的文档、程序及其数据的综合。该综合中的每一个元素均称为该软件产品的软件配置中的一个配置项。

软件配置管理和软件版本控制是紧密相关的,版本控制是实施配置管理的基础。版本控制涉及了单个配置项的版本管理和软件配置的版本管理。

软件配置管理一般包括如下步骤:①标识 识别软件产品的结构、产品的构件及其类型,为其分配唯一的标识符,并以某种形式提供对它们的存取;②控制 通过建立产品基线,控制在整个软件生存

周期中对软件产品的修改和发布；③状态统计　记录并报告软件产品和修改请求的状态；④审计　确认产品的完整性，并维护一致性；⑤发布与提交　确保发布和提交产品的正确性及其发布信息。

一般而言，由于软件系统构造和演化的复杂性，有效的软件配置管理需要一套实用的管理规则，并结合相应的配置管理工具来实施。

参考文献

1. ISO/IEC 12207:1995 Information Technology-Software-Part1: Software Life-Cycle Process. 1995

2. Burrows C, George G, Dart S. Configuration management. Ovum Ltd., 1996

（谢冰）

ruanjian sheji moshi

软件设计模式（software design pattern）　对给定的应用环境中典型软件设计问题经过多次检验的解决方案的描述。大多数模式都只在确定的环境中工作，因此必须指定模式用于何处以及如何使用。设计模式这个术语是由 Erich Gamma，Richard Helm，Ralph Johnson，John Vlissides 四人于 1995 年提出的，用以捕获富有经验的开发者的设计解决方案，而这些设计解决方案已在许多项目中得到检验。使用模式可以复用其他开发者的成功经验，开发者能在以后的开发活动中反复应用模式。模式不只是局限于设计，也适用于包括系统分析、软件开发过程本身在内的许多领域。

一个好的模式通常应该做到以下几点：

（1）它通过解决方案解答问题；

（2）其解决方案经过多次实践的检验；

（3）其解决方案通常不是显而易见，而是需要洞察力；

（4）它描述多重结构中的（复杂）关系；

（5）它服务于一个有用的目的。

描述模式的元素一般可包含以下信息：

名字　捕获模式主要特性的描述性短语；

问题　模式适用的典型问题的本质；

应用环境　模式何时使用，如何使用（例如模式最适合应用的环境）；

可应用性　模式的问题描述与现实环境匹配时的约束；

例子　实际使用模式来解决一些现实问题，有时附带实际的代码；

样板代码　在特定应用环境下模式实现的例子；

基本原理　关于该解决方案为何适用于所描述的问题的简短解释；

相关模式　它们可以共享应用环境或约束，也可以联合使用。

描述设计模式有多种变形，但是上述列表包括了多数人认同的本质要素。

Erich Gamma 等四人总结了面向对象软件的设计经验，根据两条准则对面向对象软件的设计模式进行了分类。一条是目的准则，即模式是用来完成什么工作的。根据此准则，模式可分为创建型、结构型和行为型。创建型模式与对象的创建有关，结构型模式处理类或对象的组合，行为型模式描述类或对象如何交互以及如何分配职责。另一条是范围准则，即模式主要用于类还是对象。类模式处理类和子类之间的关系，这些关系通过继承建立，是静态的。对象模式处理对象之间的关系，这些关系在运行时可以改变，具有动态性。

参考文献

1. Marciniak J J. Encyclopedia of software engineering. John Wiley & Sons Inc., 2002

2. Gamma E, Helm R, Johnson R, et al. Design patterns: elements of reusable object-oriented software. Addison Wesley, 1995

（钱乐秋）

ruanjian shengcun zhouqi

软件生存周期（software life cycle）　软件产品或软件系统从产生、投入使用到被淘汰的全过程。

在计算机技术发展的初期，人们把软件开发简单地理解为编写程序。随着软件复杂性的增长，人们认识到软件开发活动应划分为需求分析、设计、实现、测试等若干个活动，并将这些活动以适当的方式分配到不同的阶段中去完成。不同的软件开发模型之间的区别在于将这些活动分配到不同的阶段中的方式不同。

早期，人们认识到一个软件系统生存周期也和人的一生相类似，可以划分为若干个互相区别而又彼此联系的阶段，上一阶段的结果为下一阶段的依据，上一阶段没有完成决不进行下一阶段的工作，从而形成了最早的软件生存周期概念。

通常把软件生存周期分为 5 个阶段，即需求、设计、实现（编码）、测试和维护。需求包括问题分析和需求分析，问题分析获取需求定义（又称需求规约），需求分析生成功能规约；设计包括概要设计和详细设计，概要设计建立整个软件体系结构，包括子

系统、模块以及相关层次的说明、每一模块的接口定义;详细设计产生程序员可用的模块说明,即数据结构说明及加工描述;实现是把设计结果转换为可执行的程序代码;测试包括单元测试、组装测试和确认测试,这些测试活动的目的就是使软件系统达到需求时提出的各项要求;维护是对投入运行的软件进行修改,使软件系统能适应外界环境的变化、实现功能扩充和质量改善。

随着对软件生产率和软件质量的要求不断提高,人们又认识到关键在于软件开发和维护中的管理问题,认为其关键是"**软件过程**",为此进行了一系列的研究和讨论。IEEE 标准化委员会于 1991 年 9 月制定出"软件生存周期过程开发标准",接着 ISO/IEC 于 1995 年制定出"信息技术——软件生存周期过程"标准(ISO/IEC 12207:1995)。(朱三元)

ruanjian tixi jiegou

软件体系结构(software architecture) 软件总体结构的抽象表示,或以此为研究对象的学科。软件体系结构具有如下几种含义。

规定性含义

软件体系结构由结构元集、结构形以及结构理三部分组成,即

软件体系结构 = (结构元集,结构形,结构理)

其中,结构元集为一组构成软件的结构元。结构元有三类,即处理元、信息元和连接元。处理元为对信息元施行处理的构件,信息元为处理元的处理对象,连接元负责构件间的连接。结构形包括特性、联系以及权重。特性用以约束结构元的选取,联系则约束结构元间的交互与组织,权重表示特性及联系的重要程度。结构理刻画体系结构人员选取体系结构风格、结构元、结构形的动因与根据。

体系结构风格是各种特定体系结构中结构元与结构形的抽象,它不如特定体系结构约束严格,亦不如特定体系结构完备。例如,有分布式风格,多进程风格等,它们强调的只是特定体系结构的某些方面。

描述性含义

软件体系结构由构件集、连件集、模式以及约束集四部分组成,即

软件体系结构 = (构件集,连件集,模式,约束集)

其中,构件集表示构成软件的一组组成元素,连件集为一组连件,用以刻画各构件间的交互,模式为软件设计风格的描述,反映由构件及连件构成软件的构成原则,约束集中的约束表示对模式所加的限制条件。例如,在客户—服务器系统中,客户与服务器均为构件,构件间交互的描述(如过程调用、事件广播等)为连件,客户—服务器模式为模式,具体系统中对模式所加条件为约束。

多视面含义

软件体系结构为软件的一个或多个结构,每一结构反映一种视面,即

软件体系结构 = 结构集

结构 = (构件集,外部可见特性集,联系集)

其中,构件集表示构成软件的一组组成元素,外部可见特性反映为其他构件可利用该构件所作的假定,联系用以沟通相关构件。

由于软件体系结构可有多个结构,从而可有多类构件、多种联系,故在定义中并不指明何类构件与何种联系。常用的结构类型有模块结构、进程结构和概念结构等。常用的视面有代码视面、模块视面、执行视面以及概念视面。其中惯常理解的软件体系结构反映了概念视面。

学科含义

以前述各种含义的软件体系结构为研究对象的学科或谓在研究与开发前述各种含义的软件体系结构中所涉及的理论、原则、方法、技术所形成的学科。

软件体系结构发展不久,迄今未见被普遍接受的单一定义,然而,它对软件的后续开发过程以及产品质量的影响举足轻重,已成为**软件工程**的重要研究方面,且其重要性将与日俱增。

参考文献

1. Perry D E, Wolf A L. Foundations for the study of software architecture. ACM SIGSOFT Software Engineering Notes, 1992,17(4)

2. Base L, et al. Software architecture in practice. Addison-Wesley Publishing Company, 1997

3. Shaw M, Garlan D. Software architecture: perspective on an emerging discipline. Prentice Hall, 1996

(徐家福)

ruanjian tiaoshi

软件调试(software debugging) 发现所编写软件中的错误,确定错误的位置并加以排除,使之能由计算机或相关软件正确理解与执行的方法与过程。

软件错误可分为两种。由于表达方式不符合语法规则而产生的错误称为"语法错误"。它除了通过人工方式发现以外,通常可以由编译程序或者

特定的语法检查工具检查出来。排除了语法错误的软件所表达的意义还有可能与编写者心目中的想法不一致,这种错误称为“语义错误”。它往往要通过仔细阅读源程序,或者通过“让计算机实际运行或解释该软件,并观察其实际效果”的办法才能发现。

一般来说,在进行调试工作以前,首先要发现存在着某种错误的迹象。随后的调试过程通常分为两步:①确定问题的性质并且找到该错误在软件中所处的位置;②修正这一错误。在这两步工作中,第一步的工作量最大,一旦确定了错误的性质和所在位置,排除它相对来说是比较容易的。不过要注意在修正一个错误时,要防止产生新的错误。

(周锡令)

ruanjian weihu

软件维护(software maintenance) 软件产品在交付之后,为改正错误、改进性能或其他属性,或者为适应变化了的环境而对其进行修改的活动。近年来,人们更倾向于有些软件维护活动应在软件交付前就开始。

在软件交付后的整个运行期间都可能发生软件的维护活动,所以,在整个**软件生存周期**中,软件维护阶段的时间通常要比软件开发阶段的时间长得多。同时,软件维护需要对现有的软件进行修改,而这种修改可能会影响到软件中未被修改的部分。因此,在整个软件生存周期的总成本中维护的代价是昂贵的。通常,维护的成本约占生存周期总成本的三分之二,而软件开发的成本约占三分之一。

软件维护大致可分为如下四类:

(1) 改正性维护——为改正发现的问题而修改软件的活动。

(2) 适应性维护——为使其在一个已变化的或将要变化的环境中保持可用而修改软件的活动。

(3) 完善性维护——为提高其性能或易维护性而修改软件的活动。

(4) 预防性维护——为在故障发生前检测并改正软件产品中隐藏的故障而修改软件的活动。

通常把适应性维护和完善性维护归为增强活动,把改正性维护和预防性维护归为纠正活动。增强性维护主要是由于需求和环境的变更而引起的,预防性维护以预防问题的发生为目的,它经常用于对安全来说是至关重要的软件产品。

在所有的维护活动中,改正性维护约占20%,适应性维护约占25%,完善性维护占50%以上,预防性维护不足5%。由此可见,纠正性维护活动只占所有维护活动的一小部分,大部分的维护活动都是增强性维护。

多年来已经出现过多种软件维护模型,这些模型大体上都包含三个阶段:理解软件、修改软件和重新验证软件。有些维护模型包括如下阶段:确定维护目标,理解软件,产生软件变更、说明波及效应以及进行回归测试。

ISO/IEC(1999a)中软件维护主要包括如下活动。其中过程实现活动在软件交付前进行,其他活动在软件交付后进行:

(1) 过程实现活动 该活动制定维护过程中使用的计划和步骤,建立所需的配置管理组织的接口。

(2) 问题和修改分析活动 该活动评估修改请求以理解问题所在,提出解决方案,获得对实现指定解决方案的批准。

(3) 修改实现活动 该活动对软件产品实施修改并进行测试。

(4) 维护复审和验收活动 软件经修改后必须进行复审和验收,以确保软件产品的正确性。

(5) 软件移植活动 当软件产品需要从旧的操作环境转移到新的操作环境上时,需进行软件移植活动。

(6) 软件退役活动 当已不再需要某软件,或者已开发出可取代该软件的新软件时,该软件必须按一种有序的方式解除它所提供的服务,实现该软件的退役。

参考文献

Marciniak J J. Encyclopedia of software engineering. John Wiley & Sons Inc., 2002

(钱乐秋)

ruanjian xitong

软件系统(software systems) 计算机系统中由软件组成的系统。按照计算机软件的分类,软件系统分为系统软件、支撑软件和应用软件三类,系统软件主要有**操作系统**和**语言处理系统**,支撑软件通常包括**数据库管理系统**、分布式计算环境和**分布计算中间件**,**应用软件系统**是特定领域专用的软件系统,包括各种科学计算系统、数据处理系统和信息服务系统。

发展过程

自1946年诞生第一台数字电子计算机后的十多年间,计算机用户直接使用机器语言编制应用程

序,并通过控制台开关来调试和操纵程序的运行。除了应用程序外,可以说,计算机没有什么软件系统。

20 世纪 50 年代后期起,计算机运算速度和存储容量的快速增长,为软件的发展奠定了物质基础。在此期间,先后出现了 FORTRAN 和 ALGOL 等程序设计语言及其相应的编译程序,60 年代又大量出现了对计算机硬件和软件进行管理的程序,例如,美国 IBM 360 系列计算机系统的初级控制程序和英国 1900 系列计算机的执行程序等,统称为**管理程序**,属早期的操作系统,于是也有了软件的术语。70 年代以后,随着计算机应用的拓广和数据处理的发展,有效支撑数据共享的**数据库系统**应运而生。关系数据库技术及其相关理论的研究,使得数据库管理系统开始实用化、商品化。同时,操作系统、语言处理系统有了突破性进展,UNIX 操作系统,以及 FORTRAN,COBOL,C,PASCAL 等语言及其编译系统得到广泛应用。

80 年代开始,随着计算机网络的出现,分布式计算机系统成为人们关注的热点,出现了一批支持网络应用的分布式计算环境和**分布式软件系统**,并于 80 年代末 90 年代初诞生了分布计算中间件,有力支持了松耦合的分布式软件系统的发展。随着个人计算机和网络的广泛应用,软件系统开始摆脱对计算机硬件系统的依赖,从独立形态的软件商品到规模化生产的软件产业,出现了一批崭露头角的软件著名企业,如以生产个人计算机操作系统而闻名的微软公司、以生产数据库管理系统而闻名的 ORACLE 公司等。90 年代,软件产业的蓬勃发展,为整个计算机系统,乃至信息产业和现代服务业的发展注入了新的活力。进入新世纪以来,互联网被商用化激发而逐渐深入到人类社会和人们日常生活的各个角落,基于互联网的应用软件系统,以各种信息服务系统为代表,正在推动着信息社会的不断发展。

基本内容

操作系统是最靠近计算机硬件的一层软件,也是整个软件系统的核心,因而称为**系统软件**。它把硬件裸机改造成为一台更加完善的虚拟机器,使得计算机系统的资源利用率更高,运行环境更好,使用和管理更加方便。操作系统的功能通常包括管理系统资源、控制程序执行、改善人机交互以及为其他软件提供支持等几方面,分别由处理器管理、存储管理、文件管理、设备管理和作业管理等实现,其主要研究内容涉及操作系统的结构、进程(任务)调度、同步机制、死锁防止、内存分配、设备分配、并行机制、容错和恢复机制等。

语言处理系统是对软件语言进行处理的程序系统,它把用户用软件语言书写的源程序转换成能够为计算机识别和运行的目标程序,以获得预期结果。语言处理系统与操作系统均属于系统软件。语言处理系统通常有高级语言的编译程序,低级语言的汇编程序以及解释程序和各种链接编辑程序等,其研究内容主要包括程序语言的翻译技术和翻译程序的构造方法与工具,还涉及正文编辑技术、链接编辑技术和装入技术等。

数据库系统是储存、管理、处理和维护数据的软件系统。其中,**数据库**是长驻在计算机中有组织的、大量的、可共享的数据集合,数据之间的关系用数据模式来定义;数据库管理系统用于建立、使用和维护数据库。数据库系统的主要研究内容包括:数据库设计,数据模式,数据定义和操纵语言,数据完整性和相容性,数据库恢复与容错、死锁控制和防止,数据安全性等。

分布计算中间件是随着互联网的发展而兴起的一种支撑软件,主要用于支持网络环境中分布式应用软件系统的有效开发、部署、运行和管理。其主要功能是支持异构环境中的网络互联、数据整合、应用集成、流程协同和服务共享,主要研究内容包括分布式计算环境,**分布式计算模型**,**分布式计算使能技术**以及各种自适应的柔性软件技术。

人机交互系统旨在使计算机能够以友善、自然和直观的方式与用户进行信息交换或提供服务,它可以视为操作系统人机交互功能的扩展。人机交互系统与计算机图形学、人类工程学、社会心理学等学科有密切关系,其主要研究内容包括人机交互系统模型、设计原理、设计方法和系统评估等。

应用软件系统是建立在系统软件和支撑软件之上具有领域专用性的软件系统。由于它与应用领域和应用需求密切相关,故种类繁多。目前,在应用软件系统中,高性能科学计算、大规模数据处理和泛在化信息服务三类系统齐头并进,而计算机的主流应用当属信息服务。应用软件系统具有以需求为导向、以计算为核心、以网络为基础,以安全为保障的特点,其开发、运行与维护的方法参见面向领域的软件工程。

展 望

随着云计算和物联网等信息网络技术和软件服务工程的迅速发展,网络计算平台将成为软件系统

开发、部署、运行和服务的主流平台,系统软件、支撑软件和应用软件将呈现相互渗透的趋势。近年来,基于无线通信的移动设备如智能手机、平板电脑的发展,普适化的移动软件系统正在快速发展。软件系统将加快向网络化、服务化、泛在化和智能化方向演进。

参考文献

1. 徐家福. 软件技术漫谈. 计算机科学,1992,19(1)

2. Freeman P. Software perspective: the system is the message. Reading. MA: Addison-Wesley, 1987

(吴泉源　谢立)

ruanjian yanhua

软件演化(software evolution)　软件产品交付使用并上线运行后,为了适应用户需求或环境变化,对其进行维护的渐进有序的活动。

软件演化源于软件维护(参见**软件维护**)。传统的软件维护活动,主要针对运行于相对封闭环境内的软件产品,且更多关注对原有产品代码级别出现的错误(bug)修改或力度相对较小的改进(minor enhancements)。这种软件维护活动,既可在软件交付前进行,也可在软件交付后进行。软件演化则体现出一些新的特点。首先,近年来,软件产品的规模不断增大,结构和行为也更加复杂,这意味着很多错误在产品交付之前难以预知,必须在上线运行之后才能发现并进行处理。其二,软件正在从以产品为中心的制造业向以用户为中心的服务业转变,"软件即服务"已经成为软件产品的主要交付方式,这要求不断调整软件产品的功能和质量,以适应多样的用户需求。其三,当软件运行于相对开放的环境时,往往在线同时服务于多个用户,这就意味着软件产品一经交付使用后,不能完全停机。软件演化的目的,就是让软件具备持续适应快速变化的运行环境和多样化用户需求的能力,具体体现在软件构成单元数目的可变性、结构连接的可调节性、交互行为的动态适应性。因此,和传统软件维护相比,软件演化针对的软件规模更大也更复杂,对原有软件修改和调整的力度也更大(甚至可能会导致全新的软件形态);而且,软件演化往往是一个迭代的过程,需要根据用户和系统的反馈持续地在线进行。

软件演化是针对已开发并上线运行的软件产品而言。从软件生存周期角度,可将软件演化划分为三个阶段:

(1) 演化预备阶段　经过需求、设计、编码、测试、部署等软件过程,开发出具有某类功能和质量的软件产品并上线运行。

(2) 演化阶段　当用户的需求或软件运行环境发生变化时,当前的软件产品不能提供相应的服务甚至会发生错误,此时需对软件产品进行适应性调整,形成演化版本。该阶段中,演化过程通常持续迭代地进行,直至满足所有的功能或质量需求。

(3) 下线阶段　该软件不再提供服务,或者已开发出可取代该软件的新软件,此时不再需要对其进行演化,并将其下线。

从上述软件演化的几个阶段来看,软件演化同样也需要经历分析、设计、实现、测试、部署,而传统软件工程中的软件维护含义已经难以准确刻画这些活动。实际上,软件演化正在作为一个独立的软件过程,从软件维护中独立出来,成为一个极具潜力的研究方向。

参考文献

Bennett K, Rajlich V, Bennett K H. Software maintenance and evolution: a roadmap. In: Proc. of The Future of Software Engineering (FOSE 2000). 2000

(黄罡)

ruanjian yuyan

软件语言(software language)　用以书写计算机软件的语言。它主要包括**需求定义语言**、**功能性语言**、**设计性语言**、**程序设计语言**以及**文档语言**等。

需求定义语言用以书写**软件需求定义**,软件需求定义是软件功能需求和非功能需求的定义性描述。软件功能需求刻画软件"做什么",软件非功能需求刻画诸如功能性限制、设计限制、环境描述、数据与通信规程以及项目管理等。需求定义语言经历了从非形式的自然语言到半形式的语言及形式语言的发展,迄今半形式的需求定义语言已有较大进展,已逐步用于软件工程实践。

功能性语言用以书写软件**功能规约**,软件功能规约是软件功能的严格而完整的陈述。软件功能规约通常只刻画软件系统"做什么"的外部功能,而不涉及系统"如何做"的内部算法,因此,功能性语言通常又称为**功能规约语言**。从形式程度看,有非形式的功能性语言和形式的**功能语言**之分。前者是指未加限定的自然语言,后者则指其语法和语义均显式和精确定义的语言。从理论基础看,有代数类功

能性语言和逻辑类功能性语言之分。前者指以异调代数、范畴论等代数理论为主要理论基础的规约语言,后者则指以一阶谓词演算等逻辑理论为主要理论基础的规约语言。功能语言涉及规约对象、规约方法以及规约性质等。规约对象主要包括过程抽象和数据抽象两类:过程抽象是指从输入值集到输出值集的映射,其定义域和值域均由数据抽象刻画。数据抽象则提供了数据值集及其上的运算符集。规约方法涉及如何对过程抽象与数据抽象进行规约。目前,功能性语言的研究进展较大,理论基础日趋成熟,已逐步用于软件工程实践。

设计性语言用以书写软件**设计规约**。软件设计规约是软件设计的严格而完整的陈述。一方面,它是软件功能规约的算法性的细化,刻画了软件"如何做"的内部算法;另一方面,它又是软件实现的依据。从细化程度来看,有总体设计规约与详细设计规约之分。前者刻画设计的总体架构;后者刻画详尽实现细节。这两种设计规约一般都是用形式体系刻画的,亦即,都是形式的。设计性语言的发展已相对成熟,并已用于软件工程实践。

实现性语言,即一般的**程序设计语言**,用以书写计算机程序,而计算机程序是计算任务的处理对象和处理规则的描述。任何以计算机为处理工具的任务都是计算任务。处理对象是数据或信息,处理规则反映处理动作和步骤。程序设计语言的基本成分是:①数据成分,②运算成分,③控制成分,④传输成分。程序设计语言有多种分类法,例如,按语言级别分,有**低级语言**与**高级语言**,按应用范围分,有通用语言与专用语言。按成分性质分,有顺序语言、并发语言、并行语言、分布语言。按使用方式分,有交互式语言与非交互式语言等。程序设计语言的发展已有近50多年的历史,已相当成熟。

文档语言用以书写软件文档。前述的软件需求定义、软件功能规约、软件设计规约等都是软件文档。此外,还可能有一些其他的阐明性资料,以便于读者理解相应软件,它们都是软件文档。这些阐明性资料一般都是用自然语言或者半形式的语言书写的。 (徐家福)

ruanjian zaigongcheng

软件再工程(software reengineering) 通过**软件逆向工程**、重构和正向工程,将现有的软件系统重新构建,以适应新的需求的工程。广义上说,任何可以改进人们对软件的理解和改进软件本身的活动都是软件再工程的内容,如图1所示。

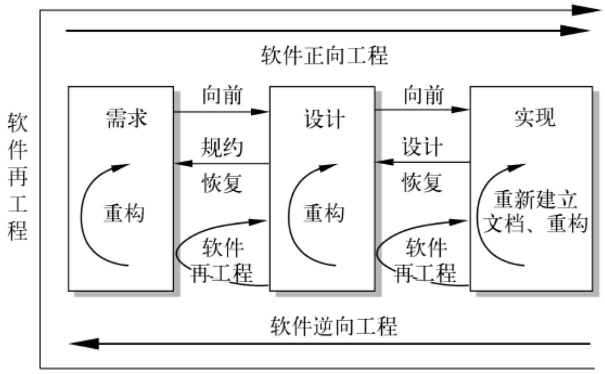


图1 软件再工程定义

由于软件再工程复用了已有的软件资源,因而往往可以以更少的开销、更短的时间、更低的风险把软件系统改造成为一个新的系统,从而在操作、系统能力、功能、性能或易维护性和可支持性上得到改进。

软件再工程的基本框架如图2所示,它给出了进行一次软件再工程实践的全过程中所应考虑的活动。但这样的框架所给出的并不是一个需要严格遵守的活动规范,针对不同的具体系统,可能需要进行相应的修改才能适应具体问题的需求。

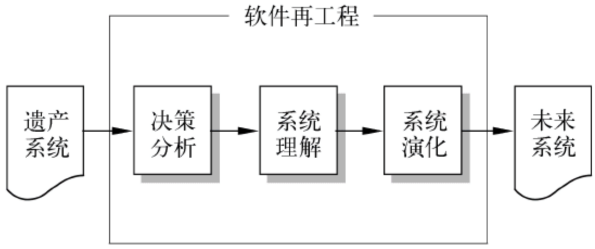


图2 软件再工程的基本框架

决策分析 决策分析的活动包括:对原有系统的详细审查和分析,明确对于改造后的新系统的需求,确定软件再工程活动的范围和方向等。决策的重要目的是在软件再工程、维护和新开发之间进行选择,最终决定是否要进行软件再工程。

系统理解 如果决策分析的结论是应该进行软件再工程,就要开始对原有的系统进行系统理解。目前,所谓的遗产系统积累得越来越多,这类系统的大部分文档和设计信息都由于时间久远而丢失,因此,我们需要通过对原有系统的源代码、设计记录以及其他文档资源的分析,才能得到原有系统的全面而详细的信息,为下一步的转化工作提供坚实基础。

系统演化 通过改造和重组，将原有系统转化为新的系统。对于一个大的系统，针对其不同部分和所期望的新系统之间的距离，可以使用不同的演化策略，主要有维护、改造和替换三种方法。

参考文献

1. Arnold R S. Software reengineering. IEEE Computer Society Press，1993

2. Feiler P H. Reengineering：an engineering problem（CMU/SEI-93-SR-5，ADA 267117）. Pittsburgh，PA：Software Engineering Center，CMU，July 1993

（孙家骕）

ruanjian zhiliang

软件质量（software quality） 反映软件系统或软件产品满足明确或隐含需求能力的特性的总和。其含义有四：其一，能满足给定需要的特性的全体；其二，具有所期望的各种属性的组合的程度；其三，顾客或用户觉得能满足其综合期望的程度；其四，软件的组合特性，它确定软件在使用中将满足顾客预期要求的程度。简言之，软件质量是软件一些特性的组合，它仅依赖软件的本身。

对于软件质量有三种不同的视面，用户主要感兴趣的是如何使用软件，软件性能和使用软件的效果。所以他们关心的是：①是否具有所需要的功能；②可靠程度如何；③效率如何；④使用是否方便；⑤移植到另一环境是否容易。开发者负责生产出满足质量要求的软件，所以他们对中间产品的质量以及最终产品质量都感兴趣，他们关心的是：①与功能和性能需求的一致性；②与开发标准的一致性；③与同行业的所有软件应满足的隐含特性的一致性。当然开发者视面还必须体现软件维护者所需要的质量特性。对于管理者也许要注重总的质量而不是某一特性，他还须从管理准则，如在进度拖延或成本超支与质量的提高之间进行权衡，以达到用有限的成本、人力和时间使质量达到优化的目的。

软件质量可以用下列特性来评价：

（1）功能性 当软件在指定条件下使用时，软件产品提供满足明确和隐含需求的功能的能力。

（2）可靠性 在指定条件下使用时，软件产品维持规定的性能级别的能力。

（3）易用性 在指定条件下使用时，软件产品被理解、学习、使用和吸引用户的能力。

（4）效率 在规定条件下，相对于所用资源的数量，软件产品可提供适当性能的能力。

（5）易维护性 软件产品可被修改的能力。修改可能包括修正、改进或软件对环境、需求和功能规格说明变化的适应。

（6）易移植性 软件产品从一种环境迁移到另外一种环境的能力。

软件质量度量 能被用来确定软件系统或软件产品某特性值的一种定量尺度。最初由 Rubey 和 Hartwick 于 1968 年提出一些属性的度量方法，但未建立质量度量模型。Boehm 等人于 1976 年提出了定量地评价软件质量的概念，提出了 60 个质量度量公式，并首次提出了软件质量度量的层次模型。1978 年 Walters 和 McCall 提出了从软件质量要素、准则到度量的三层次式的软件质量度量模型，将质量要素降到 11 个。国际标准化组织于 1985 年建议的质量度量模型是：高层——软件质量需求评价准则；中层——软件质量设计评价准则；低层——软件质量度量评价准则的三层次模型。高层由 8 个元素组成。上海计算机软件技术开发中心于 1988 年提出了软件质量评价体系，即如前述的 6 个软件质量特性，选用 22 个评价准则以及每一度量由若干度量元组成的度量族，并给出了评价准则与质量特性的关系和应用策略。国际标准化组织（ISO）和国际电工委员会（IEC）于 1991 年提出了标准，给出了 6 个特性和 21 个子特性；又于 2001 年再次修订了标准，建立了软件产品质量的两部分模型：①内部质量（软件产品在特定条件下使用时，软件产品的一组静态属性满足明确和隐含需要的能力）和外部质量（系统在特定条件下使用时，软件产品使得系统的行为能满足明确和隐含需要的能力）；②使用质量（在特定的使用周境中，软件产品使得特定用户在达到有效性、生产率、安全性和满意度等方面的特定目标的能力）。模型的第一部分为内部质量和外部质量规定了如前所述六个特性，它们可进一步细分为子特性。模型的第二部分规定了四个使用质量的特性，但没有在低于特性的层次上详细阐述使用质量的模型。

软件质量评价过程用前述软件质量特性来评价软件质量的主要步骤：一般由确立评价需求、规定评价、设计评价和执行评价四步组成。软件质量评价对于不同的角色有不同的软件质量评价过程，一般有开发者用的软件质量评价过程、需方用的软件质量评价过程和评价者用的软件质量评价过程。

参考文献

1. 朱三元，等. 软件质量及其评价技术. 北京：

清华大学出版社,1990

2. ISO/IEC 9126—2001 Software Engineering: Product Quality. 2001

3. ISO/IEC 14598-1: 1999. 信息技术软件产品评价 第1部分:概述. 1999 (朱三元)

ruanjian zhuti

软件主体(software agent) 封装的软件实体。其基本特性如下:

自主性 能在无外界直接干预的情况下独立运行;

反应性 能主动而有选择地观察环境,及时采取动作,适应环境变化;

(本原)主动性 具有表现目标制导行为的能力,必要时采取主动;

交往性 具有和其他软件主体交往之能力,一般较为主动。

软件主体可在开放、动态和多变的环境下完成指定的任务。其基本工作方式是获取环境中的信息或感知环境的动态变化,选取合适的行为方式,与其他软件主体协同与合作,对外部环境加以反应和作用,相应对内部有关机制适应调整。

面向对象技术起过且正在起着重要作用,但自20世纪80年代以来,从开放、动态和多变环境的角度考察,面向对象技术呈现出某些不足,主要表现在:自主性局限、反应性被动、封装性固定等。从而,出现了软件主体技术。对象和软件主体相比较,**对象**的自主性较为局限,在接收外界发送的消息后,即采取相应的动作,软件主体则对外界请求执行与否,自身有决定权,故其自主性较强;对象的封装性固定,其边界对环境全开放,软件主体只部分开放;对象对环境之反应被动,软件主体则能主动观察环境变化,做出反应。此外,对象缺乏目标制导行为之能力,软件主体则有这种能力。

近年来,随着万维网平台的快速发展,人们迫切需要新的软件基础技术来应对万维网平台的开放、动态和多变的特征,为软件开发平台、运行平台和开放式网络应用提供有效的基础技术支撑,从而使软件主体技术成为软件新技术研究的主流之一。很多政府机构、民间团体、大学研究机构和公司参与了软件主体技术的研究,著名的有 FIPA, UMG, CLB-FATE, USDARPA 和 EUAgentlink, CMU, MIT, Microsoft, IBM 等。

参考文献

Wooldridge M, Ciancarini P. Agent-oriented software engineering: The state of the art. In: Ciancarini P, Wooldridge M, eds. Agent-Oriented Software Engineering, LNCS 1957. Springer Verlag, 2001:1-28

(吕建)

ruanjian zidonghua fangfa

软件自动化方法(software automation method) 尽可能借助计算机系统实现软件开发的方法。计算机系统除泛指一般系统外,尤指用于软件开发的系统,特别是软件自动化系统。“尽可能”一词反映软件开发的自动化程度。软件开发是指除维护阶段外的软件生存全期,即从非形式的软件需求定义,经形式的软件功能规约、软件设计规约到可执行的程序代码、调试及至确认、交付使用的全过程。

软件自动化也可狭义地理解为,从形式的软件功能规约到可执行的程序代码这一过程的自动化。可执行的程序代码既可指低级语言程序代码,也可指高级语言程序代码。此外,自动化的程度是相对的,其高低一般因系统而异。

软件自动化一词几乎和软件一词同时出现,原称自动程序设计。早在20世纪50年代,程序人员从程序设计实践中深感程序设计工作的烦琐、不易、低效,便试图在可能范围内将一些机械性工作委交机器本身去做。在那时,实现高级语言的编译就是自动程序设计,如1956年美国国际商业机器公司(IBM)建立的第一个实用的 FORTRAN 编译程序曾被称为自动程序设计系统。60年代,出现了编译程序的编译程序和各种自编译程序。软件工程出现后,软件自动化的含义得到较大发展,其自动化的内容涉及软件生存全期的各个阶段。

软件自动化分三个不同层次:一为低级自动化:自动化系统只起程序人员的作用,亦即,从软件设计规约到可执行的程序代码这一过程的自动化。二为中级自动化:自动化系统除了起程序人员的作用外,还起设计人员的作用,甚至起部分系统分析人员的作用。亦即,从形式的软件功能规约、到设计规约、直到可执行的程序代码这一过程的自动化。三为高级自动化:自动化系统除了起程序人员、软件设计人员、系统分析人员的作用外,还起部分领域专家的作用。亦即,从非形式的软件需求定义,经形式的软件

功能规约、软件设计规约,直到可执行的程序代码这一全过程的自动化。

基本内容

主要涉及软件开发、**软件规约**、**自动生成**和**自动验证**等。

软件开发 从软件的需求定义,经形式的软件功能规约、设计规约、到可执行的程序代码,及至确认、交付使用的全过程。亦即,维护阶段除外的软件生存全期。它涉及开发风范和开发模型。开发风范指的是,软件开发的准则与风格。如自顶向下的功能分解风范,自底向上的面向对象风范等。开发模型指的是,软件开发风范的结构体现。如功能分解风范中的**瀑布模型**,面向对象风范中的**喷泉模型**等。开发模型由开发方法和工具来体现。方法和工具是一个问题的两个方面。方法有全局性方法和局部性方法两类。工具有需求分析工具、设计工具、实现工具、测试验证工具等多种。

软件规约 软件所应满足的要求的陈述。它是软件开发的依据,也是软件自动化的依据。根据陈述的性质与层次,又可区分为需求规约(即需求定义)、**功能规约**和**设计规约**等。它们分别用相应的软件语言来描述。目前,需求规约尚难以完全用形式体系刻画,一般借助自然语言。功能规约与设计规约可分别采用形式的功能性语言和设计性语言来刻画,习惯上称之为**形式功能规约**和**形式设计规约**。软件自动化要求软件规约和软件语言应有利于机器的自动处理,易于实现从软件规约自动或半自动地导出正确的程序。

自动生成 从需求规约或功能规约(取决于软件自动化的不同层次),由相应的软件自动化系统"自动"生成可执行的程序代码。这项工作的难度颇大,迄今由需求规约过渡到功能规约,还难以自动生成,尚须借助"人与系统"的交互。由功能规约到设计规约,原则上可以自动进行,而实际上,仍需借助"交互"。至于由设计规约(指详细设计规约,而不是概要设计规约)到可执行的程序代码则可自动进行,而且技术已相对成熟。软件自动生成的难度与软件自动化的程度有关,软件自动化系统的自动化程度越高,难度越大。另一方面,针对某一应用领域而设计的专用软件自动化系统实现较易;相反,通用软件自动系统则实现较难。自动生成是软件自动化系统的核心,也是困难的关键所在。

自动验证 软件的正确性是相对其功能规约而言的。凡是具备且仅具备相应功能规约中所列出的功能的软件就是正确的软件。自动验证包括三方面的工作:第一,功能规约的正确性的自动验证。功能规约的正确性是针对需求规约而言的,但由于目前需求规约还难以完全用形式体系刻画,其自动验证尚处于探索阶段。第二,可执行的程序代码的正确性的自动验证。第三,从功能规约到可执行的程序代码转换过程的正确性的自动验证。理论上说,在肯定功能规约正确的前提下,只需验证"转换过程"的正确性,作为其转换出的结果,可执行的程序代码也就必然正确。自动验证是软件自动化系统必不可少的内容,但真正实用的验证方法仍不多见。

实现方法

主要有演绎综合、程序转换、归纳综合和过程实现等方法。

演绎综合 从给定的软件规约借助演绎推理综合出程序的方法。它将数学中的构造性证明与软件开发相联系,将开发步骤解释为证明步骤,从证明中抽取相应的程序。

代表性的工作是 Z. Manna 和 R. Waldinger 的 DEDALUS 演绎算法综合系统,它从用数理逻辑形式刻画的程序输入输出规约,即前置断言和后置断言,用演绎的方法自动导出等价的 LISP 程序。

程序转换 由一程序转换至满足其功能要求的另一程序的方法。程序转换有两类:一类是纵向转换,即由一抽象级别较高的程序转至另一满足其功能要求的抽象级别较低的程序。另一类是横向转换,即在相同(或类似)抽象级别上程序间的转换。

纵向转换的代表性工作是 70 年代中期西德慕尼黑技术大学信息学研究所在 F. L. Bauer 教授主持下开始研究的软件自动化系统,即计算机辅助、直觉指导的程序设计(CIP)项目。CIP 的目标是开发可形式保证程序正确性的程序开发系统。

横向转换的代表性工作是英国爱丁堡大学 R. M. Burstall 和 J. Darlington 等人研制的 POP-2 系统和在此基础上发展起来的 ZAP 系统。这类系统根据一组规则基本自动地将作用式递归程序转换成高效的循环程序。

归纳综合 借助反映程序性态的实例,利用归纳推理导出程序的方法。其方式主要有两种:一种是"输入输出对"法:借助给出的一组输入输出对,逐步推导出适于一类问题的程序。另一种是"部分轨迹"法:通过所给实例的运行轨迹,逐步导出程序。

过程实现 借助过程化手段将一软件规约转化成目标代码的方法。在对应软件规约中的各个成分,其转换目标的相应成分明确,而且相应的转换映射也明确的前提下,该映射可借助过程来实现。

现状与展望

三十多年来,虽然软件自动化的含义不断深化,工作不断提高,但受目前计算机科学技术及有关学科发展的限制,现状还远远不能令人满意。现有多数系统均为实验性的,很少臻于实用,通用软件自动化系统更是如此。其状况表现为:第一,虽然有关软件开发风范与模型的工作不少,但从软件自动化角度,何种开发风范与模型更为合适,却未见系统性的研究工作。第二,软件规约语言的研究工作众多,包括设计性规约语言,功能性规约语言,广谱性规约语言。而需求性规约语言很少,且多半带有自然语言成分。第三,自动生成是软件自动化的核心与关键。问题是:①从功能规约生成相应的设计规约还难以达到完全自动化,尚需借助人与系统的交互;②从需求规约生成相应的功能规约的多数系统基本上仍然停留在手工阶段。第四,自动验证是软件自动化必要内容。迄今,理论上虽然可以验证生成过程的正确性,但在实际软件开发过程中却很难验证。

为使软件自动化系统达到便于使用、功能较强和功效较高等目标,需进一步开展的研究有:①语言,特别是自然语言。由于人与系统的接口宜用自然语言描述,故必须研究自然语言理解与处理。为此,一方面,要研究自然语言的形式化;另一方面,也要研究形式语言的自然化。否则,不仅软件自动化的推广应用难以实现,而且社会信息化也将流于空谈。②人工智能,特别是机器学习。研究机器学习,如果不将机器学习的原理与技术融于软件自动化的研究中,就不可能使系统能具备自适应、自学习以及自组织的能力。③数理逻辑,特别是动态逻辑。古典数理逻辑宜于描述静态系统,但是,为了描述软件生成,必须研究动态逻辑,借以奠定软件自动化的理论基础。

软件自动化是解决软件功能不强,质量欠佳和生产率低三大问题的新技术,是提高软件生产率的根本途径。随着计算机科学技术及有关学科的发展和软件自动化技术的自身完善,它在未来的软件开发过程中必将起重要作用。

参考文献

1. 徐家福,陈道蓄,吕建,等. 软件自动化. 北京:清华大学出版社,1994

2. Balzer R A. 15 year perspective on automatic programming. IEEE Software Engineering, 1985, 11(11):1257-1268

(徐家福 王志坚 张家重)

S

sanwang ronghe

三网融合(tri-network convergence)　指面向单一服务领域的传统电信网、广播电视网、互联网在向下一代电信网、下一代广播电视网、下一代互联网演进过程中,业务领域相互渗透交叉,核心网传输与交换技术趋同,分组承载和接入技术泛化,信息资源可以通过IP服务互联共享,各自能为用户提供语音、数据和广播电视等多种基础业务。

下一代电信网(next-generation network)遵循承载与控制分离的思想,采用开放、标准体系结构,能够提供话音、视频、数据等多媒体综合业务。国际电信联盟(ITU)对下一代电信网的基本特征进行了定义,下一代电信网汇聚了固定、移动、宽带等多种网络,使用多宽带及确保服务质量的传输技术,是基于IP分组技术的网络。在网络内,与服务相关的功能不依赖于与传输相关的基础技术。下一代电信网能使用户无束缚地接入网络并能促进服务供应商的竞争,支持能对用户提供个性化和无所不至服务的广泛移动性。

下一代互联网(next-generation internet)设计的目标是为了解决现有互联网在可扩展性、安全性、可管理性方面的问题,其基本特征是采用IPv6协议、具有更大的地址空间和网络规模、提供更高的网络带宽、提供服务质量的保障、具备可管理性、实现有序的管理及有效的运营维护。

下一代广播电视网(next-generation broadcast)国际上并没有统一的定义。在中国,NGB的网络特性主要体现为有线、无线相结合的天地一体化覆盖,基于广播和分组交换融合技术构建的扁平式网络体制,采用保证服务质量的大规模汇聚接入技术,可同时支持广播、组播、双向交互和推送播存四种工作模式,家庭用户实际接入速率可达100 Mb/s。NGB的业务提供特征表现为具有开放式业务支撑架构,以云计算和透明计算模式保证业务提供的便捷性、开放性与可信度。NGB的网络延拓形态为有线无线相结合的智慧家庭网络,户内**物联网**是其内在的自然属性,支持家用电器等受控终端网络化的演进。

三网融合的趋势涉及多个层面,包括技术、终端、业务以及管理体制的融合。技术的融合是指在电信网、广播电视网、互联网在演进过程中,在保证专有业务和特色技术进步的同时,核心网技术实现方式趋同,接入网络技术泛化应用,信息资源通过IP服务可互联共享。终端的融合是指手机、电视机、计算机等不同终端的功能互相交叠,能力逐渐趋同,可实现资源的共享和业务的互通,终端的网络属性多元、多模化。业务的融合指用户可以通过单一网络即可享用话音、数据和视频等多种业务。不同网络的业务允许双向进入,实现信息内容共享,并通过业务的合成,衍生出更为丰富的增值业务。管理体制的融合是指电信网、广播电视网、互联网的行业管制和政策方面也逐渐趋向统一,形成完善的法律体系,保证网络和业务的对等开放,规范运营,有效监管。

早在20世纪90年代起欧洲、美国等发达国家先后通过法律,允许电信网、广播电视网、互联网等行业有条件地相互准入或单向准入,有力地推动了互联网技术和业务的发展。中国三网融合的发展历程经历了以下几个阶段:2001年3月,《国民经济和社会发展第十个五年规划纲要》中提到“促进电信、电视、计算机三网融合”。这是“三网融合”第一次在国家的五年规划中被明确提出。2006年3月,《国民经济和社会发展第十一个五年规划纲要》再次强调,要“积极推进三网融合”。2010年1月,国务院发布《推进三网融合总体方案》,全面推进三网融合战略,促进电信网、互联网和广播电视网的融合。

参考文献

1. 国务院文件. 推进三网融合总体方案. 北京: 国发[2010]5号文, 2010

2. 曾剑秋. 网和天下——三网融合理论、实验与信息安全. 北京: 北京邮电大学出版社, 2010

3. NGB总体专家委员会. 中国下一代广播电视网(NGB)自主创新发展战略研究报告. 北京: 国家广播电影电视总局, 2010

(邬江兴)

sanwei donghua

三维动画(3D animation)　生成三维空间中场

景及形体随时间而演变的一系列可供实时播放的画面的过程和技术。三维动画是**计算机动画**的一个主要分支。在三维动画中，三维模型由计算机构造，并通过对模型、虚拟摄像机、虚拟光源和物体材料的运动控制，由真实感图形绘制算法自动生成一系列逼真的连续动态图像，故亦称为模型动画。

在三维计算机动画中，三维场景主要由 3 种类型的实体组成：物体、摄像机、光源。每种物体都有一些特征，这些特征可按照一定的规律随时间而演变。使用这些特征可模拟物体的运动、变化等。例如，对物体而言，其特征有位置（位置随时间的改变可模拟物体的平移）、方位（方位的变化可模拟机器人手臂的旋转运动）、大小（大小的变化可模拟植物生长的演变等）、形状（形状的变化可模拟云彩的变化、心脏的运动、物体的变形等）、颜色（颜色的变化可模拟火焰、日出等）、透明度（模拟雾的效果等），等等；对于摄像机，其特征有视点位置、摄像机朝向等；对光源，其特征有光源位置、强度、衰减系数、类型（点光源、线光源、面光源或体光源）等；物体材料属性包括漫反射系数、镜面反射系数、自发光系数、会聚指数、透明度、反射系数、折射率、各种纹理等。

三维动画主要研究物体的运动控制技术及与三维动画有关的造型、绘制等技术。三维动画技术大致可分为**关键帧动画**、渐变和**变形动画**、**过程动画**、关节动画和**人体动画**、**基于物理的动画**等。

参考文献

1. Parent R. Computer animation: algorithms and techniques. 2nd ed. San Fransisco, CA: Morgan Kaufmann, 2007

2. Kerlow I V. The Art of 3D Computer Animation and Effects. 4th ed. Wiley Publishing, 2009

（王裕国　金小刚）

sanwei wangge chuli

三维网格处理（3D mesh processing）　处理多边形网格特别是三角形网格模型的技术。它是**数字几何处理**（DGP）的重要内容。随着三维几何数据的获取技术逐渐成熟，研究的焦点主要集中在狭义的 DGP 技术，即基于网格的几何信息采样与规整技术。现有的主流算法框架往往与重网格化问题密切相关，可分为 5 大类：①基于半规整或者称为具有细分拓扑连接性的网格方法；②基于松弛算子的方法；③拓扑映射的方法；④流形上的标量-向量场方法；⑤流形上的 Voronoi 划分。

基于半规整网格的方法是以小波理论和多分辨率分析为基础的，如细分小波和球面小波。其基本思想是把一个复杂的函数分解为一个简单的低分辨率部分与若干相继的小波系数部分的叠加。因从三维重建算法和各种造型软件得到的网格模型通常不具备细分连接性，为此研究者提出了一系列对网格重新网格化的方法，其主要步骤包括数据分片、参数化和重采样。目前已经实现的基于细分网格技术的典型 DGP 技术应用包括多分辨率编辑、几何压缩和网格变形等。

基于松弛算子的方法代表性工作是基于信号处理的光顺造型算法和基于非均匀松弛算子多分辨率信号处理框架。前者从网格的局部邻接信息中导出了一种几何信号的离散傅里叶变换，并在其相应频域中设计带通滤波器或称为松弛算子，达到网格光顺的目的。后者的贡献在于设计了一种依赖于几何和拓扑连接的非均匀松弛算子。利用该算子，结合渐进网格的多分辨率表示，可以设计出适用于任意网格的多分辨率变换，用于网格的光顺、增强、编辑和纹理映射。

拓扑映射方法先将网格变换到简单的定义域中，然后在新的定义域中实施信号分析。如几何图像方法，将网格自动剖切为开网格，然后参数化到矩形区域上，所有网格的信号，如顶点位置、法向和颜色等，都可以均匀采样成图像。类似地，可以将网格映射到球面上，利用球面调和分析，得到完整的分析和处理效果。

大多数多边形网格是一个二维流形网格曲面，即在任意局部都与二维圆盘同胚。为此可在这样一个二维流形上定义一个或多个标量场。利用标量场上奇异点的性质，可进一步指导网格的剖分，然后实施局部参数化，进而对网格曲面实施全局重采样。换句话说，完成了重网格化操作。此类方法还可较容易地扩展到三维体网格分析与重网格化。

部分研究者观察到网格整体质量，即网格顶点的连接度，与流形上采样点的均匀程度密切相关。为此，出现了类似直接研究网格模型分布顶点，然后通过 Voronoi 图与 Delaunay 三角化的对偶关系获得整体分析结果。区别于传统松弛法中的局部优化模型，此类方法为了获得整体优化，对于所构造的质量函数采用牛顿法或者拟牛顿法等加速优化求解。由于 Voronoi 图可视为对于流形上度量分析的结果，因此对于后续的网格光顺、增强等应用带来了便利。

三维网格处理的另一个重要内容是对网格的编

码和压缩。由于应用的不同，研究者除了采用以上方法进行编码和压缩方法外，也针对顶点间拓扑连接考虑合理编码，如入度编码技术等。某些编码可形成多边形条带，以利于图形硬件进行快速绘制。

随着**网格曲面**研究的深入发展，人们逐渐意识到处理四边网格曲面的重要性。四边网格因其具有良好的对称性，非常适合于部分需要有限元计算和动画仿真的应用，从而引发了四边网格曲面重网格化以及参数化等方面的研究发展。其基本原理与三角网格有相似之处，而处理的难点在于对称性的不同，需要引入额外的约束来获得合适的网格采样结果。

参考文献

1. Lee A, Sweldens W, Schröder P, Cowsar L, Dobkin D. MAPS: Multiresolution adaptive parameterization of surfaces. In: Proceedings of ACM SIGGRAPH 1998, 95-104

2. Taubin G. A signal processing approach to fair surface design. In: Proceedings of ACM SIGGRAPH 1995, 351-358

3. Gu X F, Gortler S J, Hoppe H. Geometry images. In: Proceedings of ACM SIGGRAPH 2002, 355-361

4. Zhang M Y, Huang J, Liu X G, Bao H J. A wave-based anisotropic quadrangulation method. In: Proceedings of ACM SIGGRAPH 2010, ACM Trans on Graphics, 2010, 29(4) （鲍虎军 张宏鑫）

sanwei wangge qumian

三维网格曲面（3D mesh surfaces） 一种高效的三维曲面表示形式。该曲面形式通过顶点位置离散采样和顶点间的边相互连接，形成网状结构。**网格曲面**上的边形成多边形表面，构成采样曲面的逼近表示。根据形成网格曲面上的多边形面的种类，可分为三角网格、四边网格以及一般多边形网格曲面。三角网格曲面上每个面均是三角形，四边网格曲面每个面均是四边形面，除此之外一般统称为多边形网格曲面。在一个网格曲面中，如果每个顶点处的邻接多边形面能够展开拓扑同胚于一个圆盘，则称为流形网格曲面。流形网格曲面可用于描述现实世界中大多数曲面情况，所以是三维网格曲面处理方法的主要作用对象。三维网格曲面具有数学表达简单、易于存储和绘制等特点，适于表达复杂形体，因而在虚拟仿真、图形显示、游戏等应用中广为应用。

三维网格曲面表示可追溯到**计算机图形学**发展的早期。基于三维网格曲面表示，容易计算其逼近曲面的局部几何量，例如法向、面积、曲率等，因而可方便地实现三维物体的着色绘制（shading）。此外，网格曲面的点、边、面等拓扑元素可以附着各类绘制相关信息，例如纹理坐标等，极大地丰富了网格曲面的表现能力。值得指出的是，当代图形硬件架构以及绘制管线均以三维网格曲面的表达为核心来设计，这使得三维网格曲面的绘制极其高效。三维网格曲面是三维扫描技术的主要输出数据格式，也是**参数曲面**到图形硬件绘制过程中的中介模式，所以它在整个**数字几何处理**中占据了中心地位。

为了能在三维网格曲面粘贴纹理图像，研究者通过分析网格曲面展平过程中的各种形变度量，引出了各种**网格曲面参数化**与展开技术。针对网格曲面数据量大、顶点采样密的特点，借鉴信号处理等相关技术，开发了网格曲面简化、传输与压缩等技术。为了实现复杂物体的形状控制与修改，探讨了网格曲面整体与局部几何之间的关系，提出了高效的三维网格变形与编辑方法。由于三维网格曲面大量地用于三维扫描和逆向工程，这使得数据去噪、修补、重网格化以及曲面拟合成为核心研究内容。

随着各类大型三维网格曲面数据库的建立，以及动漫、CAD 领域中高效三维建模需求的推动，使得三维网格曲面高层信息的识别与理解成为一个发展方向。例如，基于特征或者语义的三维网格曲面分割方法，对称性探测技术，基于三维网格曲面相似性的数据检索技术，都已被纳入到了三维网格曲面的研究范畴。

参考文献

Mario Botsch, Mark Pauly, Leif Kobbelt, Pierre Alliez, Bruno Levy. Geometric Modeling Based on Polygonal Meshes. SIGGRAPH 2007, 2007 （张宏鑫）

saomiaoyi

扫描仪（scanner） 利用光电技术和数字处理技术以扫描方式将图形或图像信息转换为数字信号的装置。扫描仪在计算机中的应用始于 1984 年。扫描仪由扫描头、控制电路和机械部件等组成。扫描头由光源、光敏元件和光学镜头等组成。工作时照射到原稿（即扫描对象）上的光经棱镜和分色镜分成红绿蓝（RGB）三色各自独立的光信号。光电转换元件在接收光信号时将连续的一条线的图像分解

成许多分离的点，形成一个个像素，同时根据光线强弱不同转换成幅度不同的电信号，再经过**模数转换器**转换成 RGB 三路独立的数字信号。扫描完一行后，步进电机和机械部件使扫描头（或原稿）移动一小段距离，扫描下一行。这样一行一行地扫完原稿。得到的数字信号按扫描仪专用的 Twain 软件处理后送入计算机。大多数扫描仪采用通用串行总线（USB）接口。

扫描精度是扫描仪的一个重要指标。它是指扫描仪的光学分辨率，主要是由镜头的质量和光电耦合器件（CCD）的数量决定。由于受制造工艺的限制，CCD 扫描头的最高分辨率为 20 000 个像素（2010 年）。应用在 A4 幅面的扫描仪上，可实现 2400 dpi（点数每英寸）的扫描精度。

扫描速度是扫描仪的另一个重要指标，一般指从打开扫描仪预热，到完成图像处理的整个过程。扫描速度与系统配置、扫描分辨率设置、扫描尺寸、放大倍率等有密切关系。

扫描仪种类很多，可以按不同的标准来分类。按图像类型分有黑白、灰度和彩色扫描仪。按扫描对象幅面大小分可分为小幅面的手持式扫描仪、中等幅面（A4）的台式扫描仪和大幅面的工程图扫描仪。按扫描对象的材料分有扫描纸质材料的反射式扫描仪和扫描透明胶片材料的透射式扫描仪。按扫描仪用途分，除了通用的扫描仪外，还有专用的扫描仪，如卡片扫描仪、条码扫描仪等。

扫描仪的光电转换元件有三种：

（1）光电耦合器件（CCD）　多数平板式扫描仪使用光电耦合器件为光电转换元件。光源通常采用长条状白色发光二极管或冷阴极管，也有彩色扫描仪采用黄绿色发光二极管的。使用的光学成像系统有两种：缩小扫描型光学成像系统和等倍扫描型光学成像系统。缩小型光学系统成像采用 2.5 cm 长度的线性 CCD 作为光学系统中的图像传感器，由于 CCD 的尺寸远不及扫描原稿的宽度，这种成像系统中，在 CCD 的前面有一个镜头，用于在扫描时将原稿图像通过镜头缩小后投射到线性 CCD 上。等倍扫描型光学成像系统则采用与扫描原稿宽度相等的线性 CCD 作为图像传感器；这种光学成像系统中采用了一种特殊的镜头，它由上下排列整齐的两排棒状镜头组成；这种棒状镜头的直径为 1 mm，长约 6 mm，每一列都有 100 个以上这样的镜头阵列构成，这种成像系统在手持式扫描仪中较为常见。

（2）接触式图像传感器 CIS　CIS 感光器件使用硫化镉光敏电阻。它使用 LED 发光二极管阵列作为光源，光源照到扫描对象后直接反射到感光元件，接收到的光信号经模数转换变成数字信号。由于各感光单元之间干扰大，严重影响清晰度，扫描精度不高。CIS 型扫描仪的优点是结构、原理和光路都简单。设计制造成本低，体积小。由于必须贴近目标识别，没有景深，不能扫描实物，只适用于扫描文稿。

（3）光电倍增管　光电倍增管是大型滚筒式扫描仪采用的光电转换元件。光电倍增管是一种电子管。在灵敏度、噪声系数、动态密度范围等指标上都超过 CCD 及 CIS 等感光器件。滚筒式扫描仪扫描图像时，将要扫描的原稿贴附在透明滚筒上，滚筒在步进电机的驱动下，高速旋转。高强度的点光源投射到原稿上逐点对原稿进行扫描，并将透射或反射的光线经由透镜、红绿蓝滤色片所构成的光路将光线送到光电倍增管放大后进行模数转换，从而获得每个像素的红绿蓝的数字信号并送入计算机。它的扫描特点是一个像素一个像素地采集光信号，扫描图像的信息还原性好。

扫描仪是图像输入系统和文字识别系统中的重要设备。可以把整幅文字、图像、照片、底片直接输入计算机，大大提高了工作效率。在办公自动化领域，扫描仪用于资料制作、资料管理、机械或其他工程图纸档案的管理等。此外，扫描仪还用于**模式识别**，如指纹识别等领域。（林兼）

shaifa

筛法（sieve method）　数论中有广泛应用的一个测试素数的初等方法。大约在公元前 250 年，古希腊数学家 Eratosthenés 根据整数的如下性质，提出求素数的方法。即，如果 $n\leqslant N$，而 n 是复合数，则 n 必为一个不大于 $\sqrt{N}$ 的素数所整除，对于一个整数 N，只要知道了不超过 $\sqrt{N}$ 的所有素数，就能求出不超过 N 的全部素数。具体做法是：列出不超过 $\sqrt{N}$ 的全体素数，设为 $2=p_1<p_2<\cdots<p_r\leqslant\sqrt{N}$，然后再列出 $2,3,\cdots,N$，在其中留下 $p_1=2$ 而把 p_1 的倍数 $2p_1$，$3p_1$，…全部划去，再留下 p_2 而把 p_2 的倍数 $2p_2$，$3p_2$，…全部划去，如此继续下去，直到最后留下 p_r 而划去 p_r 的全部倍数，剩下的整数就是不超过 N 的全部素数。例如要求出不超过 50 的全部素数，因为不超过 $\sqrt{50}<8$ 的素数是 2,3,5,7，所以在 $2,3,\cdots,50$ 中留下 2,3,5,7，依次划去 2,3,5,7 的倍数，最后留下

的整数 2,3,5,7,11,13,17,19,23,29,31,37,41,43,47 就是不超过 50 的全体素数。这就是著名的埃拉托斯特尼筛法。素数表都是根据这一方法略加变化而造出来的。

埃拉托斯特尼筛法的改进与发展,是近代解析数论的重要研究课题之一。许多古典数论问题,例如孪生素数猜想(即差为 2 的素数对有无穷多对)、哥德巴赫猜想(即大于 2 的偶数可表为两个素数之和)都可用筛法研究,并且获得了部分成果。由 Ю. В. Линник 等人发展起来的大筛法已经有所进展。现在已经知道,存在一个整数 N,使得若 p 取遍所有孪生素数,则所有它们的倒数 $1/p$ 之和小于 N;而当 x 充分大时,可以使不超过 x 的所有这种素数 p 的相应和数任意大。1966 年,我国数学家陈景润证明了:所有偶数(除 2 以外)是一个素数和另一个整数之和,而后者或者是一个素数,或者仅仅是两个素数的乘积。同时,陈景润用类似方法也证明了有无穷多个素数 p,使 $p+2$ 或者是素数,或者仅仅是两个素数的乘积。这是世界公认的筛法理论卓越的应用成果,对近代筛法的进展有重要的影响。

参考文献

华罗庚. 数论导引. 北京:科学出版社,1957

(蒋增荣)

shangxiawen wuguan wenfa

上下文无关文法(context free grammar)

形式语言理论中一种重要的变换文法,在乔姆斯基分层中称为 **2 型文法**,生成的语言称为上下文无关语言或 2 型语言,在程序设计语言的语法描述中有重要应用。

范式 上下文无关文法(CFG)可以化为两种简单的范式之一,即任一上下文无关语言(CFL)可用如下两种标准 CFG 的任意一种生成:其一是乔姆斯基范式,它的产生式均取 $A\to BC$ 或 $A\to a$ 的形式;其二是格雷巴赫范式,它的产生式均取 $A\to aBC$ 或 $A\to\alpha$ 的形式。其中 $A,B,C\in V$,是非终结符;$a\in\Sigma$,是终结符;$\alpha\in\Sigma^*$,是终结符串。

歧义性和不确定性 从文法生成语言,可有多种推导方式。例如文法 $\{S\to AB, A\to a, B\to b\}$ 可有两种推导:$S\Rightarrow AB\Rightarrow aB\Rightarrow ab$ 及 $S\Rightarrow AB\Rightarrow Ab\Rightarrow ab$。若每次都取最左边的非终结符进行推导,如上例中的前一种方式那样,则称为左推导。如果有两种不同的左推导推出同一结果,则称此文法是歧义的,反之是无歧义文法。对有些歧义文法,可找到一个等价的无歧义文法,生成同一个语言。不具有无歧义文法的语言称为本质歧义性语言。例如,$\{S\to A, S\to a, A\to a\}$ 是歧义的文法。$L=\{a^mb^nc^n \mid m,n\geqslant 1\}\cup\{a^mb^mc^n \mid m,n\geqslant 1\}$ 是本质歧义性语言。接受 CFL 的自动机称为**下推自动机**。确定型和不确定型下推自动机接受的语言分别称为确定型 CFL 和不确定型 CFL,前者是后者的真子集。例如,$L=\{a^nb^n \mid n\geqslant 1\}\cup\{a^nb^{2n} \mid n\geqslant 1\}$ 是一个不确定型 CFL 而不是确定型 CFL。

同态映射下的性质 对任意正整数 n,令 $\Sigma_n=\{a_1,\cdots,a_n\}$,$\Sigma_n'=\{a_1',\cdots,a_n'\}$,定义乔姆斯基变换文法 $G=(\Sigma,V,S,P)$ 为 $(\Sigma_n\cup\Sigma_n', \{S\}, S, \{S\to, S\,a_i\,S\,a_i'\,S \mid 1\leqslant i\leqslant n\})$。这个文法生成的语言称为代克集。如果把 a_i 看作开括号,把 a_i' 看作相应的闭括号,则 n 维代克集 D_n 就是由 n 种不同的括号对组成的配对序列之集合。例如,$a_1a_2a_2a_2'\,a_2'\,a_1'$ 和 $a_1a_1'a_2a_2'a_1a_1'$ 都属于 D_2。

代克集是把正则语言族扩大成上下文无关语言族的工具。对任一上下文无关语言 L,必存在两个同态映射 h_1 和 h_2,以及一个正则语言 R,使 $L=h_2[h_1^{-1}(D_2)\cap R]$,其中 D_2 是二维代克集,反之亦然。

更进一步,上下文无关语言族是包含 D_2,且在同态、逆同态和与正则语言相交三种代数运算下封闭的最小语言族。关于其他的代数封闭性质见**形式语言理论**。

文法形式和文法的相似性 在符号置换的意义下(终结符和非终结符分别置换),许多文法之间有着相似性。把一组彼此相似的文法抽象成一个更高级的形式体系,称之为文法形式。迄今,文法形式的研究主要集中在上下文无关文法上。

文法形式的具体定义是:给定无限的终结符表 Σ_∞ 和无限的非终结符表 V_∞,任取 Σ_∞ 和 V_∞ 的非空子集 Σ 和 V,按构造普通文法的方法定义一个四元组 $G=(\Sigma,V,S,P)$。在 G 确定以后,任取映射函数 φ,把 Σ 中每一元素 a 映射为 Σ_∞ 中一有限子集 $\varphi(a)$,把 V 中每一元素 A 映射为 V_∞ 中一有限子集 $\varphi(A)$,且当 $A\neq B$ 时有 $\varphi(A)\cap\varphi(B)=\varnothing$。$\varphi$ 就是所需的置换,通过它得到一个具体的文法 $\varphi(G)=[\varphi(\Sigma),\varphi(V),\varphi(S),\varphi(P)]$,其中 $\varphi(P)$ 是把 P 中所有产生式中的符号作 φ 置换后得到的一组新产生式;$\varphi(\Sigma)$,$\varphi(V)$ 和 $\varphi(S)$ 分别是 $\varphi(P)$ 中出现的终结符集、非终结符集和出发符号。

这样的 G 称为文法形式,φ 称为 G 的一个解释,

$\varphi(G)$是G的一个解释文法,被认为是相似于G。令φ遍历各种可能的解释,得到的$\varphi(G)$集合称为G的文法族,由此生成的语言集合$\mathscr{L}(\varphi(G))$称为G的文法性语言族。例如,文法形式$\{S\to aS, S\to a\}$的文法性语言族是正则语言集;$\{S\to SS, S\to a\}$的文法性语言族是上下文无关语言集。如果要求在φ置换时对$a\neq b$有$\varphi(a)\cap\varphi(b)=\varnothing$,其中$a,b$为终结符,则在上述解释、解释文法、文法族、文法性语言族等名词前皆应加上“严格”二字。

若文法形式G作为普通文法时生成的语言$L(G)$是无限集,则称G为非平凡的。此时文法性语言族$\mathscr{L}(G)$是一个满主半AFL(参见**形式语言理论**),反之不然。如满主半AFL $\mathscr{G}(\{a^nb^n \mid n\geqslant 1\})$不是一个文法性语言族。

以$\mathscr{L}_1\cdot\mathscr{L}_2$表示文法性语言族$\mathscr{L}_1$和$\mathscr{L}_2$的乘积,$\mathscr{L}_1\cap\mathscr{L}_2$表示两者之并,它们仍是文法性语言族。如果当$\mathscr{L}\subseteq\mathscr{L}_1\cdot\mathscr{L}_2$时,必有$\mathscr{L}\subseteq\mathscr{L}_1$或$\mathscr{L}\subseteq\mathscr{L}_2$成立,则称$\mathscr{L}$是素的。正则语言集和线性语言集都是素文法性语言族。任一文法性语言族$\mathscr{L}$必可唯一地分解为它的素因子乘积和:$\mathscr{L}=(\mathscr{L}_{11}\cdots\mathscr{L}_{1n_1})\cup\cdots\cup(\mathscr{L}_{m1}\cdots\mathscr{L}_{mn_m})$,其中每个$\mathscr{L}_{ij}$都是素因子。这里所说的唯一性是相对于求和运算$\cup$可交换而言的。

子文法类 可以根据不同的观点取上下文无关文法的子文法。一种观点是根据文法的外形和它们生成的语言族在代数运算下的封闭性。例如,若文法G的产生式只具有下列三种形式:$A\to\alpha B\beta, C\to\gamma$和$S\to\delta$,其中$A,B,C\in V, \alpha\beta\gamma\in\Sigma^*, \delta\in(\Sigma\cup V)^*$,且$\delta$中至多含$k$个非终结符,$S$是出发符号,且不在任何产生式的右部出现,则称$G$为$k$线性文法。1线性文法又称线性文法。所有$k$线性文法统称元线性文法。元线性语言族在求并和乘积运算下是封闭的,但在求交、求补、乘幂闭包和置换等运算下皆不封闭。从包含关系说,正则语言族真包含于线性语言族。对任一$k\geqslant 1$,k线性语言族真包含于$k+1$线性语言族,元线性语言族真包含于上下文无关语言族。例如,$L_1=\{a^ib^i \mid i\geqslant 1\}$是线性语言而不是正则语言。$L_2=\{a^{n_1}ba^{n_1}ca^{n_2}ba^{n_2}c\cdots a^{n_{k+1}}ba^{n_{k+1}}c \mid n_1\geqslant 1,\cdots,n_{k+1}\geqslant 1\}$是$k+1$线性语言而不是$k$线性语言。

由于上下文无关文法被广泛地应用于描述程序设计语言的语法,因此更重要的是从机械执行语法分解的角度取上下文无关文法的子文法,最重要的一类就是无歧义的上下文无关文法,因为无歧义性对于计算机语言的语法分解至为重要。在无歧义的上下文无关文法中最重要的子类是LR(k)文法,它只要求向前看k个符号即能作正确的自左至右语法分解。LR(k)文法能描述所有的确定型上下文无关语言,但是对于任意的$k>1$,由LR(k)文法生成的语言必可由一等价的LR(1)文法生成。LR(0)文法生成的语言类是LR(1)文法生成的语言类的真子类。

语言识别的复杂性 Cocke-Younger-Kasami三人分别独立地发现了一个算法,对任意给定的CFG和字符串α,它能识别α是否是该CFG生成的CFL中的句子。这个算法要用到乔姆斯基标准型。厄利发现了另一个具有同样识别功能的算法,但不需要把文法化成标准型。两个算法的时间复杂性都是$O(n^3)$,空间复杂性都是$O(n^2)$,其中n是α的长度。如果CFG是线性的,则可以修改上述算法,使时间复杂性降到$O(n^2)$,空间复杂性降到$O(n)$。如果CFG是无歧义的,则时间复杂性也可降到$O(n^2)$。利用改进的厄利算法可以证明,LR(k)文法的语法分解复杂性在时间和空间上都是$O(n)$。范利扬进一步把任意CFG的CFL识别时间复杂性降到$O(n^{2.81})$。格雷巴赫发现了一个“最难识别”的CFL,其含义是:如果有一个算法能在时间$O(f(n))$和空间$O(g(n))$内识别此语言,其中$f(n)\geqslant n, g(n)\geqslant n$,则任何CFL均能在此时空限制下被识别。但迄今未能给出精确的$f(n)$和$g(n)$。

近年来开展了CFL并行识别复杂性的研究。鲁琢、李特等的算法可在$O(\log^2 n)$时间内用$O(n^6)$个处理器并行识别任意CFL。若CFL是线性的,处理器数可降到$O(n^3)$。若CFL是本质无歧义的,处理器数可降到$O(n^2)$;或者只降到$O(n^3)$,但同时使时间复杂性降到$O(\log n)$。 (陆汝钤)

shangxiawen youguan wenfa

上下文有关文法(context sensitive grammar) **形式语言理论**中的一种重要文法。一个四元组$G=(\Sigma,V,S,P)$,其中Σ是终结符的有限字母表,V是非终结符的有限字母表,$S(\in V)$是开始符号,P是生成式的有限非空集,P中的生成式都为$\alpha A\beta\to\alpha\gamma\beta$的形式,这里$A\in V, \alpha,\beta\in(\Sigma\cup V)^*$, $\gamma\in(\Sigma\cup V)^+$。上下文有关文法又称为**1型文法**。其生成式的直观意义是:在左有α,右有β的上下文中,A可以被γ所替换。上下文有关文法所生成的语言称为上下文有关语言或1型语言。常用$\mathscr{L}_1$表示1型语言类。

单调文法 若文法$G=(\Sigma,V,S,P)$的所有生成

式都为 $\alpha \to \beta$ 的形式并且 $|\alpha| \leqslant |\beta|$，其中 $\alpha \in (\Sigma \cup V)^* V(\Sigma \cup V)^*$，$\beta \in (\Sigma \cup V)^+$，则称 G 为单调文法。单调文法可简化得使 P 中任意生成式的右侧长最大为 2，即：若 $\alpha \to \beta \in P$，则 $|\beta| \leqslant 2$。已经证明：单调文法所生成的语言类与 1 型语言类，即上下文有关语言类相同。因此，有的文献把单调文法的定义作为上下文有关文法的定义。

例如：$G=(\{a,b,c\},\{S,A,B\},S,P)$，其中：$P=\{S \to aSAB/aAB, BA \to AB, aA \to ab, bA \to bb, bB \to bc, CB \to cc\}$，显然，$G$ 是单调文法，因而也是上下文有关文法。它所生成的语言 $L(G)=\{a^n b^n c^n \mid n \geqslant 1\}$ 是上下文有关语言。

上下文有关文法的标准型为：$A \to \xi, A \to BC, AB \to CD$，其中 $\xi \in (\Sigma \cup V), A, B, C, D \in V$。上下文有关语言类与线性有界自动机接受的语言类相同。1 型语言对运算的封闭性以及关于判定问题的一些结果参见**短语结构文法**中的表 1 和表 2。特别要指出的是：1 型语言对补运算是否封闭是迄今未解决的一个问题。

参考文献

Hopcroft J E, Ullman J D. Introduction to automata theory, languages and computation. Boston, MA: Addison - Wesley, 1979

（马世骅）

shebei ceshi

设备测试（device testing） 为保证可靠工作而对计算机整机系统及其各部件与子系统进行的测试。设备测试主要对计算机系统的各种设备如打印机、早期的软磁盘驱动器、**硬磁盘驱动器**以及**显示器**等的功能、机械特性、电气特性等根据事先制定的技术规范、测试标准、参数范围等进行严格的测试。测试采用专用或通用测试仪器与计量工具来实现，以保证受测设备的可靠性、互换性、稳定性和其他质量指标。

设备测试一般可分为制造过程中对关键零部件的测试及装配后对整个设备的测试。关键零部件的测试又可分为对机械部件的测试与电气部件的测试。整个设备的测试又可分为单项测试与综合测试，综合测试一般有自检和联机测试。

上述这些测试，其测试的内容、方法、所用测试仪器设备、测试的环境条件等，随设备的不同而有所不同。（参见**打印机测试**、**硬磁盘驱动器测试**）

（周学仁）

shejixing yuyan

设计性语言（design language） 用于书写软件**设计规约**的语言。设计性语言用于软件的概要设计及详细设计，其描述对象为软件系统的组织或其组成部分的内部结构，不同于重在定义软件系统及其组成部分的界面特性的**功能语言**。它也并不描述一个可以高效执行的软件的全部细节，因此有别于用于编程的**程序设计语言**。

设计性语言是软件设计人员、项目管理人员和程序实现人员之间交流的工具。一个良好的设计性语言应该包含已知最有效的控制及数据结构概念，以便简明、自然地描述软件实现策略，并能在不同的程度上省略细节，可以在不列出低层细节的条件下明确地描述算法、数据结构和子任务的基本特性。

发展过程

早期使用图形化或半图形化设计性语言，并插入用自然语言书写的正文描述。后来出现了非形式化的设计性语言，广泛地用于书写设计规约。目前结构设计中使用的仍然以非形式化的语言为主，但形式化的设计性语言受到越来越多的注意，特别是在详细设计中应用较多。

主要类型

以下简要介绍目前使用的几种主要的设计性语言，每类语言的基本成分可参见各相关条目。

图形化的设计性语言 图形化的设计描述工具有多种，广泛使用的有流程图、盒图（N-S 图）和问题分析图（PAD）等。流程图的基本成分很简单，描述方式自然，既可用于概要设计，也可用于详细设计。流程图可以支持结构程序设计，但它本身并不提供避免良好结构被破坏的机制。盒图则有此优点。盒图功能作用域明确；控制转移受严格控制；局部或全程数据的作用域容易确定；容易描述子程序递归调用。开发针对盒图的计算机辅助工具较困难，其使用受到很大限制。日本日立公司于 1979 年提出的问题分析图则特别适合于计算机支持下的详细设计。对应于不同的常用程序设计语言，可以有不同风格的 PAD 图。利用各自的对应表，能用比较机械的方法将图转换为程序。

表格形式的设计性语言 在许多应用系统中，每个模块的功能表现为对一个复合条件求值，根据结果选择适当的动作。对于这样的系统，可以用决策表作为设计描述工具。决策表早在软件工程出现之前就有人使用，随着软件工程技术的发展，现在已

有对决策表进行完备性、一致性验证和化简的工具；并且出现了直接以决策表为输入的表驱动算法。因此，决策表已经成为一种有效的过程设计语言。

设计性程序语言 这类语言书写的文档具有和实现性语言类似的结构，但控制结构内部的细节描述采用自然语言，因此，这类语言可以认为是结构化的英语，属于非形式化语言。

为了便于进行从设计到实现的转换，经常用各种高级语言作为相应设计性语言的基础，有 Ada-PDL，Pascal－PDL 等各种适合于不同开发环境的设计性语言。设计性程序语言应该具有如下特征：

（1）文法有固定的关键字，以提供描述所有结构化构造、数据说明以及模块特性的机制；

（2）一般用自然语言；

（3）可以定义简单和复合数据结构，包括抽象程度较高的数据结构，如链表、树等；

（4）可以描述子程序定义和能支持各种界面描述方式的调用。

为满足以上要求，基本设计程序语言的语法应包括：子程序定义、界面描述、数据说明和类型定义、块结构、条件构件、循环构件、I/O 构件。设计性程序语言（PDL）这个名词除指一类设计性语言外，还经常指由 S. Caine 等人提出的一种特定的设计性语言，这种 PDL 是使用最多的设计性语言之一。其格式和示例参见 PDL 语言。

按照传统的观点，设计性程序语言应当是可以扩充的；一方面通过扩充包容控制与数据结构方面的新概念，如多任务、并行处理、进程间通信、声象界面等；另一方面，通过扩充支持不同应用领域特定的结构。然而，近来的倾向是让语言有固定的结构。一则是因为 CASE 技术的发展提高了对设计过程自动支持的程度，同时也要求语言有严格的文法。因此，语言中必须包括有足够表达力的成分，能适应不断发展的应用要求，诸如用户定义抽象数据类型、类属模块、多继承机制、并行循环、非确定等待、进程的条件激活机制等。而领域特定结构则可以通过领域专用库来提供。

形式化的设计性语言 图形与表格语言中的正文部分几乎都使用自然语言，而程序设计语言本质上也仍是自然语言。因此，它们都有歧义性。要解决这个问题，应使用形式化的设计性语言。目前，在详细设计中使用代数语言等描述数据结构定义，或者在实现性语言中加入设计级的成分，而功能性语言中有的也含有设计级成分（参见**功能性语言**、**Z 语言**、**FGSPEC 语言**）。

参考文献

1. Martin J，McClur C S. Diagramming techniques for analyst and programmers. Englewood Cliffs，NJ：Prentice－Hall，1985

2. Linger R C，Mills H D，Witt B I. Structured programming —Theory and practice. Reading，MA：Addison-Wesley，1979

（陈道蓄）

sheji yanzheng

设计验证（design verification） 检验一个设计是否满足其规范要求的方法和过程。一个电子器件通常具有很多方面的特性，包括功能、性能、功耗等。相应地，设计验证也包括了很多方面的技术，例如功能验证技术、静态时序分析技术、功耗分析技术等，导致验证这个词在不同的上下文中有着不同的含义。一般来说，实现所期望的功能特性是一个电路设计需要满足的最基本的要求，因此在大多数情况下设计验证指对功能的验证。

设计验证通常可以划分为**模拟验证**和**形式验证**两个大类。模拟验证通过对电路施加一组特定的输入、分析其在模拟器上的输出，从而检查设计是否符合预定的规范。形式验证则利用数学方法来严格地证明一个芯片设计是否满足所给定的规范。

模拟验证包括 4 个主要环节：①激励生成；②模拟；③结果检查；④覆盖率评估。首先，验证工程师根据对功能规范的理解，开发用作电路输入的测试用例（称为激励），也可以基于一定的约束条件，自动生成随机的激励。第二个环节主要用软件模拟待验证电路的工作过程（参见**数字系统模拟技术**），产生对应于各个激励的电路输出。提高模拟速度（例如采用硬件加速的方法）有助于在更短时间内验证更大的功能空间。结果检查环节通常采用与参考模型的输出结果相比较的方式来实施。覆盖率评估用于分析模拟验证的质量，其中，功能覆盖率评估的效果依赖于所定义的功能覆盖点的完备性，而基于代码的覆盖率，如代码覆盖率、条件覆盖率、翻转覆盖率等，主要关注语句或电路结构被执行或被访问的概率。运用覆盖率评估方法，可分析出模拟验证的漏洞，用于指导进一步的激励生成。

形式验证大体上可分为三类：①等价性检验；②模型检验；③定理证明。等价性检验是指证明两个设计模型具有相同的功能。模型检验则是证明一个设计是否满足某种属性。定理证明是指将待验证

电路的性质表示成某形式化系统中的公式,如果该公式能从形式化系统的公理和推理规则中推导出来,那么设计的电路就具有该性质。定理证明要求用户深刻理解基本逻辑与形式证明,在学术领域之外的实际应用还很少见。等价性检验和模型检验已集成到商业工具中,在芯片设计中得到广泛应用。

随着设计规模和复杂度的增长,设计验证在芯片设计流程中越来越重要。模拟验证作为工业界主流的验证技术,在面对大规模的复杂设计时,其不完备性越来越突出。形式验证得到学术界和工业界越来越多的重视。二者结合之后又产生了半形式化验证,试图融合模拟验证在处理规模方面的优势和形式验证的完备性,成为设计验证的一个重要发展趋势。

参考文献

1. 李晓维,吕涛,李华伟,李光辉. 数字集成电路设计验证——量化评估、激励生成、形式化验证. 北京:科学出版社,2010

2. 韩俊刚,杜慧敏. 数字硬件的形式化验证. 北京:北京大学出版社,2001

（李晓维）

shehui jisuan

社会计算(social computing) 借助计算机技术研究社会活动、社会结构、社会组织和社会功能的理论和方法。社会计算属于计算机科学与社会科学相交叉的研究领域。它基于社会科学知识、理论和方法学,借助计算技术和信息技术,帮助人类认识和研究社会科学以及信息技术在社会中的应用等各种问题,提升人类社会活动的效益和水平。

社会计算的产生与发展主要有两方面的线索。一方面,从微观技术层面,20 世纪中叶,研究学者就开始提出将计算技术融入心理和组织发展学。90 年代后期,随着**互联网**发展,社会计算越来越体现出重要的应用价值,开始成为重要研究领域。另一方面,从宏观层面,20 世纪 70 年代,一些研究机构开始研究复杂性科学,并使用计算机模拟社会相互关联的繁杂网络和复杂系统。近年来,随着各种媒体和移动计算技术等的迅猛发展,大量社会数据产生,有力地推动了社会计算技术的研究。

社会计算的研究目标可概括为:

(1) 建立针对网络化社会中新型问题的社会科学理论,构建计算社会学的基础理论框架,为复杂社会问题的建模和实验提供社会科学基础;

(2) 建立社会计算中系统建模、模型验证、现实与虚拟世界的互动融合等核心理论的基础,研究社会计算中的计算和学习方法论;

(3) 构建统一的、可编程的社会计算实验平台和实验环境。从社会安全、应急管理、经济系统安全、工程安全等领域入手,研究现实社会系统中的实际问题及相应的解决方案。

社会计算的研究内容可归纳为以下两个方面:

(1) 从计算技术作用于社会活动这个角度看,社会计算是设计、实施和评估各种促进人与人之间的交流、协调和合作的信息技术。它以人和活动为中心,利用先进的信息计算技术,采用多学科交叉方法,如心理学、人类学、广告学、工程学等,通过现场观察、采访、人物情景假想等,支持人们的高度有效交流。近来,社会计算更加关注利用信息技术工具实现社会性的交互和通信,使人们能方便地构建人与人之间沟通的虚拟空间。这类改进信息技术工具以协助个人进行社会性沟通与协作的技术被称为社会软件如,Email、BBS、办公自动化系统、群件等许多传统网络工具都是社会软件。而博客、Wiki、微博等更强调借助网络工具等有效利用用户群体的智慧如,**社会网络**(social network)、社会网络服务(social networking services)、群体智慧(collective intelligence)等均属于这方面的研究。

(2) 从计算技术如何影响社会科学的学科发展角度看,社会计算将人文理论与信息系统相融合,利用计算技术解决以往社会科学研究中使用经验方法和数学方程式等手段难以解决的问题,例如,利用计算技术研究传统政治社会学问题、人文社会静态知识的动态化、定性社会讨论的数字化、社会孤立知识的网络化等,促进社会的发展和规划科学化。社会计算以传统人文社会科学理论为指导,建立一套用科学计算方法帮助解决政治经济等社会问题的理论和方法学;研究建模与计算方法,为解决新兴社会问题提供整套理论和技术支撑。这方面的研究包括社会网络分析、内容计算、社会情感分析、人工社会等。

社会计算的应用非常广泛,涉及技术层面上的企业应用、宏观层面上的社会重大问题的应用,以及教育、金融、文化等领域的社会计算问题。典型的应用领域包括:①复杂工程系统的社会计算,如人工交通、人工电网、人工制造、人工生产、人工农业系统等;②人口系统的社会计算,支持国家人口综合规划、人口政策、人口动态管理和控制;③复杂生态问题的社会计算,将人的生态观念和认识嵌入人工生

态系统的计算与决策中,评估不同生态政策对自然环境和人类社会的影响及效果;④复杂经济系统的社会计算,如人工股市、经济模型、人工能源系统等;⑤模拟战争系统中的社会计算,以人工军事系统有效逼真地模拟战争;⑥复杂社会系统的社会计算,如新兴的互联网社会群体计算等。

社会计算发展的时间较短,虽然在一些领域,已经获得了一些理论上的研究成果,但由于社会系统的复杂性,在理论和应用方面都仍然存在许多难以解决的问题。从社会计算的未来发展趋势看,网络社会媒体信息处理、动态社会群体的建模与演化、网络化社会态势计算分析与评估等将成为社会计算的重要研究方向。

参考文献

1. 钱学森, 于景元, 戴汝为. 一个科学的新领域: 开放的复杂巨系统及其方法论, 自然杂志, 1990,13(1)

2. 毛文吉,曾大军,柯冠岩,等. 社会计算研究的现状与未来. 中国计算机学会通讯,2011,(2):8-11

3. 王飞跃,李晓晨,毛文吉,等. 社会计算的基本方法与应用. 杭州:浙江大学出版社,2013

(叶允明　刘挺)

shehui wangluo

社会网络(social network)　指社会个体成员之间因为互动而形成的相对稳定的关系体系,包括了社会关系中的个体、个体间的连结以及连结上的资源等。社会网络中的个体既可以是个人,也可以是集合单位,如家庭、部门、组织等。

社会网络的要素主要包括网络结构、节点、资源、关系(联接方式)等。①节点　即社会网络中的个体,是社会网络的基本内容和构成社会网络的能动实体;不同节点以一定的关系结合起来,就形成了社会网络;②资源　是社会网络中的个体及整体所拥有的各种有价值的内容,与节点相关联;③关系　主要指节点之间的联接方式,也可指节点与资源之间的联系;④网络结构　是整个社会网络的核心组织,涉及相关领域及覆盖范围。社会网络与企业知识、信息等资源的获取密切相关;网络中的成员有差别地占有各种稀缺性资源;关系的数量、方向、密度、权重以及个体在网络中位置等因素直接影响着资源流动的方式和效率。

社会网络的发展经历了三个阶段:

(1)社会网络一般被认为起源于英国人类学。20世纪30年代,英国人类学家A. R. Brown首次使用了“社会网”概念。50年代,一些人类学家开始系统地发展网络概念。Elizabeth Bott第一次提出了网络结构的测量工具——结(knit),使得网络分析受到许多社会学家的注意。

(2)60年代开始,美国社会心理学家Moreno在分析人际关系时创立的社会测量法则为社会网络分析奠定计量分析基础。后来,随着矩阵分析方法等数学方法的引入,社会网络分析在社会学、经济学得到长足发展,从一种具体的研究方法拓展为一种理论框架。

(3)伴随着20世纪90年代后期至21世纪初**互联网**的发展,社会网络在互联网上获得了更加广泛应用,如Facebook、Twitter、Myspace、人人网、新浪微博和腾讯QQ等;大量用户利用这些互联网社会计算工具,构建了自己的人际交互关系,形成虚拟空间上的社会网络。

社会网络的重要理论基础源于“六度分隔”(six degrees of separation)理论,即指:任意两个陌生人之间所间隔的人不会超过六个。“六度分隔”说明了社会中普遍存在的“弱纽带”;通过弱纽带,人与人之间的距离变得非常相近。按照六度分割理论,每个个体的社交圈都不断放大,最后成为一个大型网络,这就是社会网络。目前,有关社会网络理论、技术和应用的研究主要是围绕着社会网络分析(social network analysis)展开的。根据社会网络中的社会关系结构化的特点,基于数学方法、图论等发展起来的社会网络分析方法成为一种新的分析工具得到了社会学家的广泛使用。社会网络分析是研究社会网络的一组行动者(如人、社区、群体、组织、国家等)的关系结构及其属性关系的一套规范和方法。人在社会环境中的相互作用可以表达为基于关系的一种模式或规则,可以反映社会结构。社会网络分析关注的焦点是关系和关系的模式,主要分析不同社会单位(个体、群体或社会)所构成的社会关系的结构及其属性。

社会网络主要有两方面应用:①传统领域的应用　包括各种政务应用、商业应用、教育应用、医疗应用等。在政务应用方面,社会网络被应用到各种政府事务中,使得政府可以快速获取公众意见,并保持公众对政府的动态联系。在商业应用方面,社会网络可以帮助企业寻求客户和潜在客户,并通过社会网络进行销售和广告活动。在医药应用方面,社

会网络可以作为医疗信息分享、推荐、发布的一个平台。②新兴**互联网**领域的应用　包括社会性网络服务(social networking services, SNS)、博客、微博等各种社会性软件,旨在帮助人们建立社会性网络的互联网应用服务,利用信息技术工具构建人人相互沟通的虚拟空间。

随着互联网的迅猛发展,社会网络的基础理论、分析方法和技术体系遇到一系列新的机遇和挑战。目前,社会网络研究的主要发展趋势为:

(1) 海量社会网络　海量社会网络分析技术用于应对当前呈爆炸式增长的社会网络节点数目的海量规模。

(2) 移动社会网络　使社会网络具有更强的实时性、动态性。

(3) 多语义社区网络　使得社会网络的社区性不再只基于节点之间的联系和通信,而是可以通过多种关系形成社会网络和社区。

参考文献

1. 林聚任. 社会网络分析. 北京:北京师范大学出版社, 2009

2. 林顿 C. 弗里曼. 社会网络分析发展史. 张文宏,刘军,王卫东,译. 北京:中国人民大学出版社, 2008

3. 刘军. 社会网络分析导论. 北京:社会科学文献出版社, 2004

(叶允明　王岢　刘挺)

shehui wangluo xitong

社会网络系统(social network systems)　用于管理、分析与挖掘社会网络的系统。

社会网络是由参与者构成的社会结构,其中参与者可以是个人、公司、社会团体等;参与者之间由一种或多种类型的关系相连接,如朋友关系、成员关系、共同兴趣关系、信息传播关系、疾病传播关系、信任关系以及贸易关系等。

社会网络通常被建模为图,其中顶点代表参与者,边代表参与者之间的关系。最简单的社会网络只包含单一类型的参与者和单一类型的关系。复杂的社会网络包含多种类型的参与者和多种类型的关系,并且关系可以是有向的。

社会网络系统是一种新兴的数据管理系统,目前尚无公认的模型、语言和理论基础,但这并没有阻止社会网络系统在应用层面的发展。国际上典型的社会网络系统包括美国的 Facebook、LinkedIn、MySpace、Twitter、Last. FM 等。由于中国国家防火墙的存在,在中国无法访问 Facebook、Twitter 等网站,因而产生了诸多替代品,如人人网、新浪微博等。

对社会网络的研究起源于 19 世纪后期,主要由社会学家进行。现在社会网络分析中使用的大多数概念(如密度,连通性)都是在 20 世纪 50 ~ 60 年代提出的。1967 年,在 Milgram 著名的小世界实验中,他要求每个参与实验的人通过其认识的人将信件传递给一个指定的目标。实验发现成功的信件传递平均需要 5 个中间人,该发现后来被称作“六度分隔”理论,该理论后来被物理学家和计算机科学家使用计算机在互联网、万维网和社会网络等大规模网络中得到验证。

在学术界,研究人员使用图论及网络分析等理论和技术研究社会网络的性质及规律,这类研究被称作社会网络分析,分为社会网络的静态分析与动态分析。社会网络静态分析研究社会网络中顶点度数的分布、社会网络的直径、社会网络的构成模型和社会网络的生成模型等;社会网络的动态分析研究社会网络中顶点度数的分布、社会网络的直径随时间变化的性质和信息在社会网络中传播的性质及规律等。

参考文献

Wasserman S, Faust K. Social network analysis. Cambridge University Press, 1994

(李建中　邹兆年)

shejiao wangluo

社交网络(social network)　个人与个人、组织与组织或个人与组织之间在社会交互(social interaction)过程中建立起来的一种相对持久稳定的网络化结构。在这种网络中,节点对应于社会主体(个人或组织),而节点之间的联系则表示主体之间的社会关系。近年来,随着 Web 2.0 的大力发展与迅速普及,社交网络更多的是指用户依赖于互联网与移动通信等信息技术手段建立起来的在线社交网络(online social network)。和传统的社交网络相比,在线社交网络具有以下几个突出的特点:①虚拟性。通常,在线社交网络中的主体可以是虚拟的,其主体往往与真实社交网络中的主体并不一一对应,主体的属性可能与相应用户的真实信息存在很大的差异。一般来说,这种特性有利于保护真实用户的权利和隐私,但也带来了网络社会的信任问题。②无界性。借助于互联网全球性的普及,在线社交网络可以把更大范围内更多更广的人群联系起来。这对

于传统的社交网络来说是很难做到的。③动态性。在线社交网络中主体的加入与退出更加灵活,这使得整个在线社交网络显示出更高的动态性。④涌现性。由于在线社交网络的无边界联络以及信息技术的加速作用,使得在线社交网络在结构、功能、事件等方面呈现出大量的突变和涌现现象。

在 Web 1.0 时代,支撑社交网络的服务平台主要包括了被广泛使用的**电子邮件**系统、网络论坛系统和**即时通信**系统(如 ICQ、QQ、MSN 等)。进入 Web 2.0 时代以后,尽管这些传统社交服务平台仍然有着广泛的应用,但也涌现出大量新型的服务平台来支持用户进行更宽泛意义下的交流与互动,从而形成了多种多样的社交网络。新型社交网络服务平台不同于传统平台的关键在于其用户档案中社会关系的公开性。也就是说,在新型平台上,一个用户的"朋友"列表对他/她的"朋友"是部分或全部公开的。这一特性使新型服务平台上社交网络的构建变得更加容易和高效。

新型社交服务平台大致可以分为以下几种,即社交网站、博客与微博客、内容分享、任务合作、虚拟人生、社会标签和社交网络集成系统。此外,还有一些平台用于其他的社交网络服务,如事件和新闻发布服务等。

社交网络形成的关键依赖于网络主体之间因为社会交互行为而形成的社会关系。在 Web 1.0 时代的电子邮件系统中,这种交互表现为人与人之间相互的邮件发送与接收行为;在网络论坛上,表现为一个用户对其他用户所发帖子的阅读、评论、回复、转载等行为;在即时通信系统中,则体现为用户基于互联网的实时对话行为。在 Web 2.0 出现之后,这种社会交互比以前具有更广泛的内涵,譬如博客和微博客上一个用户对另外一个用户的关注;内容分享平台上一个用户对其他用户共享出来的图片、音频、视频的浏览和评议;任务合作平台上用户之间的分工与合作;虚拟现实中在相关虚拟环境下的用户的合作与竞争等。

和在传统社交网络中一样,在一个社交网络中主体的行为不仅受其自身所处环境的制约,而且受其在社交网络中所处位置的影响,这种影响通过主体之间的社会交互行为来进行。社会交互存在强弱之分,可以从四个不同的角度来定义:互动频率(互动的次数多为强交互,次数少则为弱交互)、感情力量(感情较强、较深为强交互,感情不深则为弱交互)、亲密程度(关系密切为强交互,疏远则为弱交互)、互惠交换(互惠交换多而广为强交互,反之则为弱交互)。强弱不同的社会交互行为在社交网络中发挥着不同的作用,强交互维系着群体、组织内部的关系,而弱交互则在群体、组织之间建立纽带联系,也就是说弱交互充当着群体或组织之间的信息桥梁。

参考文献

1. Easley D, Kleinberg J. Networks, crowds, and markets: reasoning about a highly connected world. Cambridge, UK: Cambridge University Press, 2010

2. Scott J. 社会网络分析法. 张军,译. 重庆:重庆大学出版社, 2007

3. Schneider HL, Huber LM, eds. Social networks: development, evaluation and influence. Nova Science Publishers, 2009

4. Hillstrom L. Online social networks. Lucent Books, 2010

(程学旗)

shepin shibie biaoqian

射频识别标签(rfid tag) 射频识别系统中用于承载信息的设备。又称**电子标签**、射频标签,主要由存有识别代码的大规模集成电路芯片和耦合元件(包括线圈和天线)构成。每个标签具有唯一的电子编码。高容量射频识别标签中还含有可写入的存储空间,附着在物体上用于标识目标对象。

射频识别标签的主要特性有供电方式、发射频率和作用距离几个方面。

按照供电方式可以分为有源电子标签、无源电子标签及半无源电子标签。有源式标签又称为主动标签(active tag),标签的工作电源完全由内部电池供给;无源电子标签又称为被动标签(passive tag),所使用的电源取自天线接收到的无线电波能量;半无源电子标签内芯片的工作由电池供电,射频使用的电源取自无线电波能量。

按照发射频率可以分为:低频标签(LF tag),工作频率为 30 ~ 300 kHz,目前常用的是 125 kHz 或 134.2 kHz;高频标签(HF/RF tag),工作频率为 3 ~ 30 MHz,目前常用的是 13.56 MHz;超高频标签(UHF tag),工作频率为 300 MHz ~ 3 GHz,目前常用的是 860 ~ 960 MHz(ISO/IEC 18000—6 标准);微波标签(microwave tag),工作频率大于 3 GHz,目前常用的是 2.45 GHz 与 5.8 GHz。

无源低频标签和高频标签一般通过感应耦合(inductive coupling)方式获取能源,并通过负载调制

(load modulation)方式传输数据,作用距离为 0.1 cm ~ 1 m;无源超高频标签和微波标签一般通过后向散射耦合(backscatter coupling)方式获取能源,并通过反射截面层调制(reflection cross-section modulation)方式传输数据,作用距离为 1 ~ 10 m。各种频率的射频识别标签,在有源方式下,其作用距离与发射功率有关,最远可达到几千米。

制定电子标签的国家标准,进一步降低标签成本、提高标签识别率是射频电子标签技术迅速发展需解决的主要问题。　(崔莉)

shepin shibie yueduqi

射频识别阅读器(rfid reader)　一种能阅读(有时还可以写入)电子标签(参见**射频识别标签**)数据的自动识别装置。电子标签又称应答器。它在射频识别系统中,阅读器用于读写电子标签上存储的信息、控制读写过程,并与后台的管理系统进行数据交换。通常由耦合模块、收发模块、控制模块、接口单元(如 RS-232 和 RS- 485)和供电模块组成,常见的有手持式和固定式两种。

阅读器是射频识别系统中的主设备。它是一个射频收发器,一旦进入工作状态,会发射调幅信号来激活电子标签。如果遇上了无源型电子标签,还要给它传输电能。当然,电能的传输是受到多种外部条件限制的。比如在美国,超过 1 W 的能量就不允许经无线传输。对读卡器的性能要求是很严格的,因为它必须从所收到的各种反射信号中提取出标签所反射的微弱信号。

阅读器控制与应答器之间的通信流程,并通过接口单元将数据传输给后台应用程序进行处理。阅读器的发射频率决定了应答器的工作频率,同时阅读器通过感应耦合或反向散射耦合给无源应答器提供能量和时序,两者之间可以通过全双工、半双工和单工通信的方式(参见串行传输)进行信息交换。按照阅读器与应答器之间的最大通信距离可分为:近耦合射频识别系统(<1 cm),远耦合射频识别系统(<1 m)和长距离射频识别系统(>1 m,最远可达到几千米)。

根据应用需求,阅读器还可以采用防冲突技术、认证技术和安全技术,以同时识别并处理多个应答器,并确保数据的完整性和安全性。

作为条形码的替代产品,射频识别标签由于其抗污染等恶劣环境、远距离和多标签同时识别等优点,应用非常广泛。目前典型应用在:动物晶片、门禁控制、航空包裹识别、文档追踪管理、包裹追踪识别、物流、移动商务、产品防伪、运动计时、汽车晶片防盗器、停车场管制、生产线自动化、物料管理等领域。　(崔莉)

shexiangji biaoding

摄像机标定(camera calibration)　确定摄像机成像几何参数(称为内参数)的过程和方法。

摄像机标定是从多幅二维图像恢复场景三维几何结构必不可少的步骤,是**计算机视觉**的重要研究内容。由于摄像机制造厂家提供的出厂参数一般来说不能满足应用精度的需求,所以在具体应用中需要对使用的摄像机进行标定。摄像机标定可以分为传统标定和自标定两大类。传统标定是指利用结构已知的高精度的标定块进行标定的过程。自标定是指不需要标定块,仅仅利用多幅图像之间几何基元(如点、线等)之间的对应关系进行标定的方法。自标定理论本质上利用的是这样一个事实:射影空间的绝对二次曲线(或绝对二次曲面)在图像上的像仅与摄像机内参数有关,而与摄像机运动无关。摄像机标定一般是指对针孔成像模型下成像参数的确定过程。在精度要求很高的应用场合,需要考虑摄像机的非线性畸变参数,畸变包括径向畸变和切向畸变,一般来说,径向畸变需要首先考虑。

近年来,一些大视场成像设备得到了广泛应用。主要有鱼眼相机(fish-eye camera)和反射折射相机(catadioptric camera),这些相机的视场角大,图像畸变大,标定方法相对复杂。

参考文献

Hartley R, Zisserman A. Multiple view geometry in computer vision. 2nd ed. Cambridge University Press, 2004

(胡占义)

shenshushi yuyan

申述式语言(declarative language)　着重描述要处理什么,而非如何处理的非命令式语言。如何处理要取决于实现,后者包含直接反映在语言中的算法信息。申述式语言程序是关于问题解之约束陈述,这些约束迫使含于语言实现中的算法处理机制生成一个解或一组解。函数式(或作用式)语言与逻辑式语言均为申述式语言。

函数式语言的基本运算单位是函数,其程序是函数。函数从输入变元获取输入值,回送的结果即

为函数本身之值。函数式语言的基本成分有：基本数据对象（如整数对象）；定义新数据对象（如表）的设施；基本函数（如基本算术运算）；以及定义函数的机制。函数式语言是一种面向值的语言，无状态，因而无状态变化，无副作用。具有引用透明性，函数值只取决于变元值，具相同一组变元的函数，其值唯一。换言之，在程序中某处对一表达式求值并不影响其他表达式的求值，从而任一表达式的值由其变元值及所含常量唯一决定。

1960 年诞生的 LISP 实际上已开创了函数式语言程序设计风格。1977 年著名计算机科学家 J. Backus 在他接受 Turing 奖的演讲“程序设计能否从冯·诺依曼式的设计风格中解放出来？”中，从全新的角度深刻分析了冯·诺依曼计算机和以之为基础的命令式语言的本质缺陷，明确提出了一种纯函数式程序设计语言 FP 后，函数式语言发展迅速，例如，融进软件工程思想的 SML，ML，商用纯函数式语言Miranda以及 Hope，Pebble，Haskell 等。

逻辑式语言的基本运算单位是谓词。谓词定义了变元间之逻辑关系。以逻辑式语言 Prolog 为例，它采用一阶谓词逻辑的子集——Horn 逻辑。其程序由围绕某一主题的事实、规则和询问三类语句组成，这三类语句分别用来陈述事实、定义规则和提出问题。程序的执行是根据预先读入之事实及其关系，按询问给出回答。在逻辑式语言程序中，程序与数据统一，程序构作可以是增殖式的。

为在当前使用中仍占统治地位的冯·诺依曼体系计算机上寻求申述式语言的高效实现，不少申述式语言嵌入了若干命令式语言成分，从而多少掩盖了申述式语言的独特风格。

目前申述式语言尚处于蓬勃发展中，主要研究课题有：并行性、功效、大型软件编制以及与面向对象语言的相互渗透与有机结合。

参考文献

1. Backus J. Can programming be liberated from the von Neumann style? A functional style and its algebra of program. CACM，1978，21（8）

2. 郭浩志. 程序设计语言概论. 长沙：国防科学技术大学出版社，1989

（郭浩志　徐家福）

shenhuajia jicheng dianlu

砷化镓集成电路（GaAs integrated circuit，GaAs IC）　采用砷化镓一类半导体材料制造的集成电路。这种材料是Ⅲ-Ⅴ族化合物中最典型的一类半导体材料。

与最为常用的硅半导体集成电路相比较，砷化镓集成电路具有运算速度快、功耗低、工作电压低、输入阻抗高、反向隔离度大、集成度高、抗辐射能力强和制造的工艺流程简单等优点。砷化镓集成电路已成为超高速集成电路（VHSIC）的专用名字，是近年来最为关注的、发展最迅速的超高速集成电路，但是目前它的产量和价格还远远不及硅集成电路。用砷化镓制成的各种门电路、触发器、分频器到大规模运算电路、存储器、门阵列、数字转换电路等都已实用化。作为无线发射、接收领域的应用，例如卫星广播电视通信、蓝牙技术、无线网络、全球个人移动通信、空中及陆地交通管理等，都离不了砷化镓超高速集成电路。

由于分子束外延（MBE）、金属氧化物化学气相淀积（MOCVD）等材料的生长技术与电子束（EB）等微细加工技术的出现，诞生了 GaAs，InP 等异质结高电子迁移率器件。其中有高电子迁移率晶体管（HEMT）、伪晶高电子迁移率晶体管（PHEMT）、异质结双极晶体管（HBT）、异质结场效应晶体管（HFET）等开关器件及毫米波单片集成电路（MMIC）。砷化镓超高速集成电路的门延迟时间小于 1 ns 或工作频率大于 1 GHz。可用砷化镓门电路和触发器等构成与硅集成电路相类似的电路品种。

砷化镓超高速集成电路的逻辑形式

常用的逻辑形式有直接耦合场效应晶体管逻辑（DCFL），缓冲场效应晶体管逻辑（BFL），源耦合场效应晶体管逻辑（SCFL），肖特基二极管场效应晶体管逻辑（SDFL），低夹断电压场效应晶体管逻辑（LPFL），电荷耦合场效应晶体管逻辑（CCFL）以及互补电平场效应晶体管逻辑（CLFL）等。

直接耦合场效应晶体管逻辑（DCFL）采用增强型场效应晶体管作为开关元件，耗尽型场效应晶体管或电阻作为负载，具有单电源、低功耗、电路结构简单、易实现和集成度高的优点。这种逻辑形式在砷化镓大规模集成电路设计中最为常用。其缺点是电平摆幅小，噪声容限低，要求电路中的场效应晶体管阈值偏差要小，这对砷化镓材料制备工艺和电路制造技术提出了较高的要求。

砷化镓超高速集成电路（VHSIC）最早采用的逻辑形式是缓冲场效应晶体管逻辑（BFL）。VHSIC 使用耗尽型场效应晶体管，由双电源供电，带有输出缓冲级以实现电平移位和提高驱动能力。其优点是电路速度快，驱动能力强，输出逻辑的摆幅大，噪声容

限大，并且对电路中的晶体管阈值偏差要求比较宽松，制造工艺简单等。这种电路由双电源供电，耗尽型器件功率损耗又大，因此在中小规模集成电路中使用较多。

源耦合场效应晶体管逻辑（SCFL）可采用增强型或耗尽型场效应晶体管。增强型场效应器件功耗低，集成度高，工艺难度比 DCFL 要小。由于这种逻辑电路的输出逻辑摆幅与阈值电压关系不大，且因具有互补输出而易构成触发器以及时钟频率较高等优点，已在各种规模的砷化镓超高速集成电路中得到广泛的应用。

肖特基二极管场效应晶体管逻辑（SDFL）电路采用耗尽型场效应晶体管，并由两路电源供电，扇出和驱动能力不强，工作速度也不如上述三种电路。

互补电平场效应晶体管逻辑（CLFL）是一种新的逻辑电路。它在 DCFL 的基础上采用互补信号的输入和输出。电路模拟指出，在相同速度下其功耗可比 DCFL 降低 5 ~ 500 倍，因而适用作低功耗电路。

砷化镓超高速集成电路的逻辑元件

砷化镓超高速门电路　砷化镓超高速集成电路中最简单、最基本的电路，由它可构成各种功能的超高速集成电路。砷化镓超高速门电路的传输延迟时间达 100 ~ 500 ps，输出波形的上升和下降时间为 100 ~ 200 ps。

砷化镓超高速 D 型触发器　它通常采用 DCFL、BFL 和 SCFL 的逻辑形式，是砷化镓超高速集成电路的基本电路。它具有 50 Ω 负载的驱动能力，工作频率达 8 ~ 10 GHz，输出波形的上升和下降时间为 100 ~ 200 ps，建立和保持时间为 100 ~ 150 ps。

砷化镓超高速分频器　通常又分为静态和动态两种分频器。静态分频器由门电路和触发器构成，动态分频器也称预定标器，多由 BFL 和传输门组成。静态分频器最高频率已达 20 ~ 30 GHz。一种用 HEMT 器件采用 DCFL 电路制成的 1/8 动态分频器其工作频率达 26.5 GHz，功率为 573 mW。

砷化镓存储器电路　砷化镓超高速集成电路在向大规模集成电路发展的过程中，首先研制的是静态随机存取存储器（SRAM），且主要采用功耗较小的 DCFL 电路。原因是其成本较高，存储容量也有待加大。

砷化镓门阵列和标准单元　砷化镓门阵列产品已达万门级水平。一种 3 万门 GaAs MESFET 采用 SCFL 逻辑形式的门阵列的门延迟时间为 90 ps，功耗仅为硅 ECL 的 1/4。GaAs 高速门阵列产品近年发展很快。

砷化镓微处理器　用单片集成技术制作在 GaAs 衬底上的中央处理器与用硅材料的相比，砷化镓微处理器具有高速、低功耗的特点。

砷化镓运算器　用砷化镓材料制成的加法器和减法器与用硅材料的相比，其运算速度要快得多。例如，用 HEMT 器件制成的 16 × 16 复数乘法器用 22 000 个有源器件构成 4500 个门电路，工作频率达 520 MHz，功耗为 4 W。

砷化镓模数和数模转换器　大多采用 SCFL 逻辑形式。已有 14 位、工作频率为 2 GHz、功耗为 2.5W 的数模转换器产品。

砷化镓分布式放大器　它在带宽、性能和芯片尺寸方面有优越性。已有 5 ~ 100 GHz 的 5 级 7 dB 的砷化镓分布式放大器，单位面积增益达到 1 dB/mm^2。

砷化镓功率 MMIC　砷化镓功率 MMIC 的性能可达：12.0 ~ 16.0 GHz，2 W，18 dB；42.5 GHz，0.5 W，15.1 dB。

砷化镓毫米波 PHEMT　砷化镓毫米波 PHEMT 的性能可达：16 GHz，7 W，4.5 dB；55 GHz，219 mW，4.1 dB。

由于 MBE、MOCVD 等能生长几个分子层厚度的薄膜材料的设备与电子束、反应等离子体刻蚀（RIE）微细加工技术的出现，诞生了以Ⅲ-Ⅴ族化合物 GaAs 为主体的、只有几个分子层厚度的各种材料的薄层结构，例如 AlGaAs/GaAs，InGaAs/GaAs，InAlAs/InGaAs/InP。再加上 HEMT、HBT 和弹道式收集晶体管（BCT）的研制成功，将使 GaAs 超高速集成电路向工作速度更快、功率更大、耗损更小的方向发展。

参考文献

1. Wing O. Gallium Arsenide Digital Circuits. Boston: Kluwer Academy Publishers, 1990

2. 李效白. 砷化镓微波功率场效应晶体管及其集成电路. 北京：科学出版社，1998

（汪锁发　杨玉芬）

shenjing jisuan

神经计算（neural computing）　通过对人脑的基本单元（神经元）的建模和联结，来探索模拟人脑神经系统功能的模型，研究**人工神经网络**信息处理的本质和能力，以及研制一种具有学习、联想、记忆和**模式识别**等智能信息处理功能的人工系统。

1943 年心理学家 W. S. McCulloch 和数理逻辑学家 W. Pitts 建立的 MP 神经网络模型,开创了神经计算的研究。1969 年 M. Minsky 和 S. Papert 在"Perceptron"一书中证明了**感知机**的局限性后,神经计算一直处于低谷状态。直到 1982 年美国物理学家 J. J. Hopfield 提出了 Hopfield 神经网络模型,神经计算的研究才又活跃起来。

神经计算主要研究神经网络模型,包括网络连接的拓扑结构、神经元的特征、学习算法等。目前,已有近 50 种神经网络模型,其中有**反向传播网络**、**感知机**、**自组织映射**、**Hopfield 网络**、**玻耳兹曼机**、**自适应谐振理论**等。根据连接的拓扑结构,神经网络模型可以分为前向网络和反馈网络两种。人工神经网络所具有的适应性行为通过学习实现,即根据环境的变化,对权值进行调整,从而使系统的行为得到改善。神经网络学习算法的基础是 Hebb 学习规则,由加拿大心理学家 D. Hebb 在《行为的组织》一书中提出,学习导致突触的联系强度和传递效能的提高,即为 Hebb 学习规则。在此基础上,人们提出了各种学习规则和算法,以适应不同网络模型的需要。有效的学习算法,使得神经网络能够通过连接权值的调整,构造客观世界的内在表示,形成具有特色的信息处理方法,信息存储和处理体现在网络的连接中。

神经计算中训练时所要进行的数学运算,主要是连接权值调节量和各单元多路输入信号量的矩阵乘、加计算,而且相对独立的单元及其连接可以并行处理。因此,神经处理器或神经计算机主要有以下三种实现技术:

(1) 计算机软件仿真　通过编程,在通用计算机上实现某种学习算法(例如 BP 算法)来建立并训练一个"软"神经网络。

(2) 用专用并行处理器器件进行硬件仿真　用这样的器件组成专用协处理器将显著加速"软"神经网络的学习速度,缩短其训练时间。

(3) 用线性集成电路器件　直接把电子神经元模拟线路组成的人工神经网络制作成集成电路的神经处理器芯片。由于在技术上难以改变连接电阻之值来调整连接权值,所以要根据特定的用途,利用人工神经网络开发工具在通用计算机上训练好"软"神经网络,取得有关数据,再根据这些数据设计超大规模集成电路掩膜,制作专用芯片。

神经计算研究开辟了一条不同于符号计算的信息处理途径,尤其在涉及自动控制、模式识别、人工智能等应用方面,已取得了不少用符号处理技术难以达到同样效果的成果,包括文字识别、语音识别、目标辨识、**自然语言理解**、**专家系统**(例如,金融市场预测和观测数据回归分析等)、过程控制等。

参考文献

史忠植. 神经网络. 北京:高等教育出版社,2009

(史忠植)

shenjing jisuanji

神经计算机(neural computer)　一种在生物神经元数学模型的基础上构造人工神经网络(ANN)以实现计算和信息处理的**非传统计算机**。又称神经处理器。神经计算机的组成原理和工作方式完全不同于传统计算机。后者是模拟人类基于符号化概念的抽象思维的信息处理功能,而神经处理器则是在神经网络微观结构这一层次上模拟其对刺激-反应的自适应调整来进行信息处理。

神经计算实质上是 ANN(参见**人工神经网络**)在经过学习或训练后所表现出来的对输入信息的适应性反应。但是,ANN 毕竟是生物神经网络的高度简化模型,无论是数量上还是品质上都远远比不上生物神经网络,尤其是无法与人脑神经系统相比拟。通常是针对某一特定问题设计一个 ANN 并加以训练后,用于求解该特定问题。比较现实的研究开发途径是把神经处理器作为协处理器,在嵌入式系统或控制系统中发挥其特定作用。

训练 ANN 时所要进行的数学运算,主要是连接权值调节量和各单元多路输入信号量的矩阵乘、加计算,而且相对独立的单元及其连接可以并行处理。神经计算机主要有以下 3 种实现技术:

(1) 计算机软件仿真　通过编程,在通用计算机上实现某种学习算法来建立并训练一个"软"神经网络。

(2) 用专用并行处理器器件进行硬件仿真　用这样的器件组成专用协处理器将显著加速"软"神经网络的学习速度,缩短其训练时间。

(3) 用线性集成电路器件　直接将电子神经元模拟线路组成的 ANN 制作成集成电路的神经处理器芯片。这要根据特定的用途,利用 ANN 开发工具在通用计算机上训练好"软"神经网络,取得有关数据,再根据这些数据设计 VLSI 掩膜,制作专用芯片。由于线性放大器电路的 VLSI 芯片集成度难以提高,所以主要是针对特定用途,生产一些集成度适中的产品。

ANN 的研究开辟了一条不同于传统计算机的信息处理途径。尤其是在涉及自动控制、模式识别、人工智能等应用方面,已取得了不少用传统计算机难以达到同样效果的成果,包括文字识别、语音识别、目标辨识、自然语言理解、专家系统(例如,金融市场预测和观测数据回归分析等)及过程控制等。

参考文献

1. Wasserman P D. Neural Computing—Theory and Practice. New York: Van Nostrand Reinhold, 1989

2. 史忠植. 神经计算. 北京: 电子工业出版社,1993

3. 施鸿宝. 神经网络及其应用. 西安: 西安交通大学出版社,1993

(童频)

shengwu jisuan

生物计算(biological computing) 指利用生物组织或生物元件进行计算活动或与计算相关的其他活动(如数据存储等)。生物计算是近年来出现的一个新兴的信息科学领域,其主要目的是基于生物规律和生命科学的理论与技术发展新的计算理论;通过设计和实现新的计算设备,突破传统的计算理论及其发展中的限制和瓶颈。与单一着重于提高操作速度的现有计算技术不同,生物计算技术主要着眼于利用生物元件的高度并行性,将计算任务分解为许多极小的部分,采用巨量的计算元件和极度并行方式完成计算。生物计算的速度将比现有计算机的速度高几个数量级。

生物计算是一个崭新的研究领域。到目前为止,生物计算领域比较深入的方向主要有 DNA 计算和基于合成生物学的细胞计算。DNA 计算始于二十世纪 90 年代;最初由 L. Adelman 在 1994 年提出并应用于 Hamilton 路径问题。DNA 计算主要基于合成和分裂 DNA 元件之间的化学键完成。后来,随着合成生物学的发展,人们可以设计特定的基因、代谢通路、信号传导通路甚至全新生命种类。通过设计特定的通路,可以有效地利用细胞的代谢和信号传导解决计算问题。这些都为解决更加复杂的计算问题打下了基础。

生物计算的基本原理是基于生物分子(如 DNA、蛋白质等)。通过纳米生物技术设计特定的生物分子系统,使系统的中的生物分子可以按特定方式进行交互,实现计算活动。生物计算机由一条或一系列特定的代谢通路构成。这些通路控制针对特定计算任务设计的生物物质,以一定的系统条件作为输入,将由这些通路产生的某些生化物质作为系统输出,并以一定手段对其进行测量和解读,从而得到计算结果。目前,主要有三类生物计算机:生物化学计算机(biochemical computer)、生物力学计算机(biomechanical computer) 和生物电子计算机(bioelectronic computer)。

(1) 生物化学计算机 主要利用多个生化反应构成的反馈环实现计算功能。在生物分子网络中,基于大量生物反应构成了众多具有正/负反馈效应的反馈环;利用生物分子之间的调控机制,可以构造涉及一系列生物元件的通路,并使构造的通路在一组特定化学条件下产生一类产物,其他化学条件下产生另一类产物。将这些不同的产物作为信号,通过测量其浓度作为计算结果。

(2) 生物力学计算机 与生物化学计算机相似,以特定的化学条件作为输入,测量在不同化学条件下特定通路产生的特定产物作为输出结果。其不同点在于,生物力学计算机不是基于产物的浓度,而是基于特定产物在特定初始条件下的形态作为输出。通过测定和解读输出产物的三维结构信息,即得到了计算结果。

(3) 生物电子计算机 不同代谢产物具有不同的导电性,可以通过设计特定的通路使其在不同的化学条件下所产生的产物具有不同的导电性。生物电子计算机是通过设计特定产物,测量产物的导电性作为输出结果。由于基于电信号作为输出结果,生物电子计算机可以执行电子计算。

生物计算技术的研究刚刚起步,其基本计算模型尚处于探索阶段。当前的主要研究成果是针对某些特定但典型的计算问题(如 Hamilton 路径问题)设计可用于现有生物器件的计算方法,并通过生化反应进行实验、求解和验证。其主要目的在于研究和验证计算的基本思路和基本运算器件的可用性。目前,生物计算领域研究投入最大、发展最快的是 DNA 计算技术。DNA 计算技术利用 DNA 分子间的碱基互补特性建立计算问题编码和求解方法。由于 DNA 分子体积极小,其信息密度远远大于现有基于硅质的计算器件;同时,由于 DNA 分子间生化反应的同时性,DNA 计算具有高度的并行性。因此,虽然在计算模型上与现有计算理论相比并没有很大改变,其总体计算能力不可同日而语。DNA 计算技术将对很多现实计算问题的求解带来深远的影响。

虽然目前仍处于较为初级阶段,但生物计算理论和技术已经体现出了一些远优于现有基于硅质的

计算理论和技术的特点。生物计算的优势主要体现在五个方面。①生物计算机的能耗较现有计算技术大幅降低,以 DNA 计算为例,其发热量极小,在理想条件下,其能耗最低仅为现有计算机的十亿分之一。②生物计算的存储量巨大,同样以 DNA 存储为例,1 克 DNA 作为介质可以实现 10^{20} 字节以上级别的信息存储。③生物元件相对于现有计算机的各种元件,其更加易于获取。④生物计算可以实现极度并行运算,有利于解决规模巨大的问题。⑤生物计算为控制细胞级别的化学产物产生过程提出了可能性,通过生物计算为人类在分子层面进行可控的化学反应,创造更加有效的生产方式带来了希望。

在未来的研究中,生物计算仍将着重于两个主要的研究方向。①基于生物计算元件的通用计算模型的研究。通过对通用计算模型的研究可以突破现有计算理论和模型的瓶颈,以解决在现有计算模型下因具有较高计算复杂度而难以解决的问题。同时,可以使生物计算技术通用化,以应用于更为广阔的计算领域和更为实际的计算问题。②对生物计算的基本元件本身的研究。对基本计算元件的研究将降低生物计算模型的应用难度,使生物计算技术真正具有实用性和可用性。

参考文献

1. Adleman L M. Molecular computation of solutions to combinatorial problems. Science, 1994, 266(5187): 1021-1024

2. Lim W A. Designing customized cell signaling circuits. Nature Reviews, 2010, 11: 393-403

3. Friedland A E. Synthetic gene networks that count. Science, 2009, 324(2931): 1199-1202

4. Wikipedia. Biological computer. http:\\en.wikipedia.org/wiki/Biological_computer

(王亚东)

shengwu jisuanji

生物计算机(biocomputer) 利用脱氧核糖核酸(DNA)或蛋白质等生物分子进行存储、检索、处理数据的计算系统。生物计算机包含一系列经设计的用以表现依据输入状态进行响应的生物材料代谢途径,这些代谢响应构成计算输出。生物计算机的分子级集成密度和超大规模并行处理能力使其在运算速度、信息存储密度、能量损耗和体积等方面与传统电子计算机相比具有巨大优势。生物计算机可以分为三种主要类型:生物化学计算机、生物力学计算机和生物电子计算机。它们分别利用生物分子的化学响应、力学响应和电响应实现信息处理功能。

生物计算系统研究包括器件和系统两个方面。

利用有机(或生物)材料在分子尺度内构成有序体系,提供能通过分子层次上的物理化学过程完成信息检测、处理、传输和存储的基本单元,称为分子器件。1974 年,A. L. Aviram 和 M. A. Ratner 首次提出分子整流器模型。1978 年,F. Carter 明确提出分子器件概念。尽管生物处理器芯片研究仍处于实验室阶段,但在生物元件,特别是在生物传感器的研制方面已取得不少实际成果和应用。目前对生物处理器芯片的研究表明它的尺寸可望超越超大规模集成电路工艺的极限,而且它本身具有较高的自适应性、丰富的时变特性以及有利于大规模互连等优点。2013 年,Drew Endy 领导的 Stanford 大学生物工程组宣布,已经制造出晶体管的生物等效体。这一发明是用于建造全功能计算机的三个必要元素(数据存储、信息传输、基本逻辑系统)中的最后一个。

生物计算系统不仅对器件有新的要求,其结构和计算原理也不同于传统的计算系统。它的结构一般是并行分布式的。信息的存储往往是短时记忆(快过程)和长时记忆(慢过程)的结合,是通过学习来完成的。它的计算则表现为复杂的动态过程,不存在精确的时间同步,甚至要在分维时间尺度上才能描述。1994 年,美国南加州大学提出研制 DNA 计算机的设想,并用 DNA 计算方式求解了 7 点哈米顿路径问题,显示了 DNA 计算的巨大潜力。以色列魏兹曼科学中心于 2002 年实现了一种由酶和 DNA 组成的可编程分子计算机,并于 2004 年实现了带有输入输出组件的 DNA 计算机。

生物计算机存在各种功能性能力,目前主要包括二进制逻辑操作和数学计算。

美国麻省理工学院(MIT)人工智能实验室的 Tom Knight 首先提出一种生物计算模式,蛋白质浓度用作最终用于逻辑操作的二进制信号。生物计算机的一个化学通路中的某个生化产品的浓度等于或者超过一定级别,表明“1”和“0”中的某个信号值,低于这个级别的浓度表明“1”和“0”中的另一个信号值。使用这种方法进行计算分析,生物计算机能够执行逻辑操作,只有在初始条件满足特定的逻辑约束时,合适的二进制输出才会出现。换句话说,合适的二进制输出是从一组初始条件中推导出来的结论。

除了上述类型的逻辑操作,生物计算机也能展示其他功能性能力,比如数学计算。一个例子是,

W. L. Ditto. 于 1999 年在 Georgia Tech 建造的一台生物计算机，它由水蛭神经元组成，能够执行简单的加法。

这只是生物计算机已经工程化实现来执行的著名用途中的一些。生物计算机的能力正在变得越来越复杂。由于与生产生物分子和生物计算机相关的可行性和潜在的经济效率，生物计算机技术发展是个受欢迎而又迅速增长的研究主题，将来可能看见更多进展。

参考文献

1. Conrad M, ed. Molecular computing: the lock-key paradigm. Computer, 1992, 25(11): 6-20

2. 韦钰，甘强. 分子计算理论与神经网络. 东南大学学报，1990，20(6)：109-114

3. Biocomputer. http://en. wikipedia. org/wiki/Biocomputer. Retrieved Nov. 2014

（甘强　官伯然　张广艳）

shengwu tezheng shibie

生物特征识别(biometrics)　计算机利用人体所固有的生理特征或行为特征来进行个人身份辨认和(或)验证的过程、方法与技术。生理特征与生俱来，多为先天性的，比如指纹、脸像、虹膜、掌纹等；而行为特征则是习惯使然，多为后天性的，比如语音、唇动、手势等。

常见的生物特征识别技术主要有**指纹识别**、**人脸识别**、**虹膜识别**、**说话人识别**、**掌纹识别**、**手势识别**等。由于单个生物特征识别在鲁棒性和稳定性方面还不够完备，因此出现了将多个特征综合考虑的**多模态生物特征融合**技术。

生物特征识别技术还是和谐人机交互的重要组成部分。在和谐的人机交互环境下，计算机主动观察用户，根据用户的行为或动作趋向推断其意图，主动为其提供服务。生物特征识别技术将为计算机在非接触式的环境下主动识别用户的身份奠定坚实的基础。

发展历史

利用生物特征鉴别身份的技术历史悠久，据考证，早在公元前 7000 年～前 6000 年，指纹即已经作为身份鉴别的方法在中国和古叙利亚有所应用，比如留有匠人指纹的黏土陶器，印有起草者指纹的文件等。1880 年，科学家发现指纹的唯一性，使得指纹用来作为身份识别的基础得以确认。19 世纪 90 年代，人类学家 Alphonese Bertillion 将生物特征作为一门辨认的科学加以研究，并用来辨识罪犯的身份，后人将其称之为“Bertillonage”（贝迪永式人体测定法）。20 世纪 60 年代，Bell 实验室的 S. Pruzansky 提出了基于模式匹配和概率统计方差分析的说话人识别方法，形成了说话人自动身份识别研究的高潮。1936 年，眼科专家 Frank Burch 指出虹膜具有独特的信息，可以用于身份识别。1987 年，Aran Safir 和 Leonard Flom 提出了利用虹膜图像进行自动识别的概率，并获得专利。自 20 世纪 60 年代以来，由于计算机及相关其他技术的发展，基于生物特征的自动身份识别技术获得了迅猛的发展。

目前，生物特征识别已经进入了实际应用阶段。国外许多高技术公司正在试图用虹膜、指纹、脸像、语音等特征取代人们手中的信用卡或密码，并且已经开始在机场、银行、电子商务以及各种电子器具上进行了实际应用。国内也有多家研究机构和开发单位正在从事多种生物特征识别技术的研究与开发，指纹识别、脸像识别等技术已经趋向成熟，并逐渐走进人们的工作和生活中。

原理与特点

传统的身份识别方法，通过识别一些标识个人身份的事物来进行身份的辨认。大体上可以分成两类：①特定的持有物，如身份证、信用卡、钥匙等；②特定的知识，如用户名和密码等。然而这些传统的方法存在容易丢失或被伪造、遗忘或记错等缺点，而且这些方法无法区分真正的拥有者和取得身份标识物的冒充者，一旦他人获得了身份标识物，将可以拥有相同的权力。

与上述传统的身份识别方法相比较，基于生物特征的身份识别技术由于使用了人体所固有的生理或行为特征，因而具有：①不易遗忘或丢失；②防伪性能好，不易伪造或被盗等优点。当然生物特征识别技术也存在着如下一些缺点：①成本较高；②存在拒识和误识现象。特别是后者是目前生物特征识别所面临的主要难点。如何降低拒识率和误识率，提高识别的性能，是目前研究的目标和重点。

各种生物特征识别技术的原理大致相同，不同的只是一些具体的处理方法。图 1 为生物特征识别的基本原理框图。

生物特征识别系统的实现过程包括训练和识别两个阶段。无论是训练还是识别，都需要首先对原始的生物特征信号进行预处理，滤除掉原始信号中的不重要的信息及背景噪声等。然后进行特征提

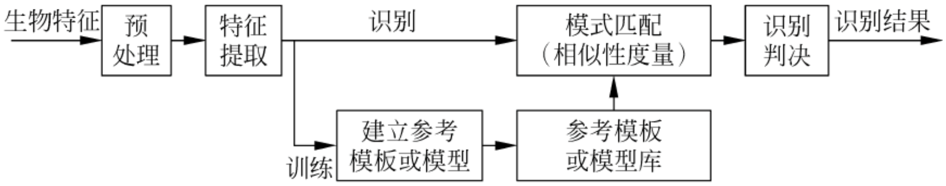


图 1　生物特征识别的基本原理框图

取，提取出反映信号特征的关键特征参数，以降低维数并便于后续处理。在训练阶段，每个用户输入若干次某种生物特征，系统经过上述预处理和特征提取后对其进行分析，据此建立每个用户的模板或模型库，或者对已在库中的该用户的模板或模型作适应性修正。由于该阶段为系统的每个用户都注册了自己的信息，所以又称之为注册阶段。在识别阶段，将待识别的用户所输入的生物特征的特征参数与在训练过程中建立的参考模板或模型加以比较，并经过一定的相似性准则进行识别判决。

分　类

生物特征识别包括辨认和确认两大范畴。

辨认　试图识别出用户的身份，是一个多选一(1∶N)的问题。系统输入某种生物特征，然后据此从库中选出与该特征相似性最优的模板或模型，并将其相似性和某个阈值加以比较，若其大于该阈值，则给出对应的用户的身份，否则，做出用户没有注册(不属于当前参考模板或模型库)的判决。可以看出辨认的过程中，系统决策的选择数目为库中所包含的参考模板或模型的数目，因而辨认的性能随着库的规模增大而降低。

确认　试图确定某个用户是否具有其所声称的身份，是一对一(1∶1)的问题。系统接受生物特征输入的同时声称该特征所对应的身份，然后系统将该特征与库中和所声称的身份对应的模板或模型进行匹配，并和某个相似性阈值加以比较，给出接受(得到确认)或拒绝(拒绝确认)的判决，因此确认的性能与库的规模无关。

性能评价

生物特征识别除基本的正确率、速度、存储容量等性能评价的标准外，还有两个重要的统计性能指标：错误拒绝率(FRR)和错误接受率(FAR)。错误拒绝是指生物特征的拥有者被系统拒绝接受；而错误接受是指将冒充者识别为真正的生物特征的拥有者。上述两个错误率考虑了识别系统的整体性能，如果单纯从算法加以考虑，主要的指标为错误非匹配率(FNMR)和错误匹配率(FMR)。

假设拥有者和冒充者的特征参数符合一定条件的分布，则系统判决以及错误拒绝率和错误接受率的示意如图 2 所示。

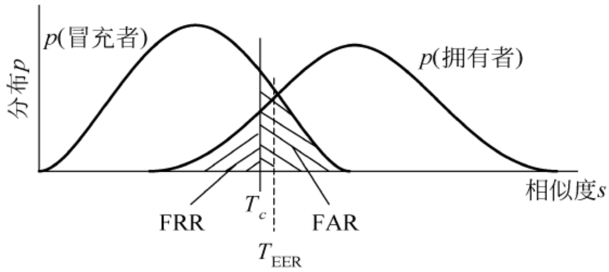


图 2　拥有者和冒充者分布以及分类错误率

对于理想的系统来说，错误拒绝率(FRR)和错误接受率(FAR)都应该是零，即拥有者和冒充者的特征分布可以无错误地加以区分。而实际系统中两个错误率是相关的，而且和所给定的用于识别判决的相似性阈值 T_c 有关。阈值越大，系统接受的相似性要求也就越大，因而更容易造成错误拒绝，即错误拒绝率 FRR 越大，而错误接受率 FAR 则越小。反之亦然。系统往往需要在两个错误率之间取一个折中。用 ROC 曲线可以很好地反映两个错误率之间的关系，如图 3 所示。曲线上的点表示在某个给定的相似性阈值下得到的错误拒绝率和错误接受率。图 3 中所标出的等错误率点(EER)表示错误拒绝率 FRR 和错误接受率 FAR 相等的点，此时的相似性阈值为 T_{EER}，在图 2 中亦有表示。有时，等错误率 EER 也可以作为指纹识别系统(参见**指纹识别**)的性能标准。

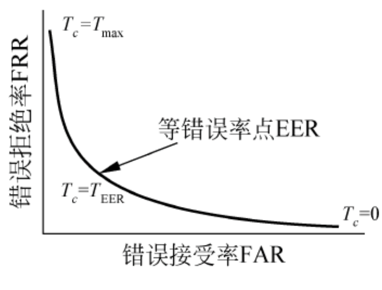


图 3　ROC 曲线

发展趋势

目前,单个生物特征识别技术中,还没有哪一种能做到完美无缺,它们在鲁棒性、稳定性等方面均存在一定的问题。比如在噪声环境中利用语音的说话人识别系统将不能很好地运行;对于双胞胎,仅仅使用脸像特征也不能很好地区分;而对于指纹识别技术,有5%左右的人不能得到很好的指纹特征等。

将多种生物特征进行融合,综合考虑其总体效果可以有效地提高系统的性能,取得更高的鲁棒性和稳定性。如脸像、语音、指纹等生物特征的融合系统,当在噪声环境中语音不能发挥很好的作用时,其他两个生物特征仍然能够得到很好的辨识效果。这正是多模态生物特征融合所研究的内容,也是今后生物特征识别领域的发展趋势。

参考文献

1. http://www.biometrics.org

2. Ratha N K, Senior A, Bolle R M. Automated biometrics. In: Proceedings of ICAPR-2001, Rio de Janiero, Brazil, 2001

3. Jain A, Hong L, Pankanti S. Biometrics: promising frontiers for emerging identification market. Comm. ACM, 91-98, February, 2000

(吴志勇　蔡莲红)

shengwu xinxi xitong

生物信息系统(bioinformatics system)　运用信息技术,获取、存储、加工和利用生物学数据,并为生物学研究与应用提供支持和服务的**应用软件系统**。

生物信息系统的研究可以追溯到20世纪60—70年代,当时出现了替换矩阵和序列比对等一些生物数据处理算法。80年代开始,生物信息系统处于成型和发展期,相继建立了以生物数据的获取、收集、存储为主要特征的世界三大核酸数据库GenBank、EMBL和DDBJ以及蛋白质数据库和人类疾病数据库等生物数据库,还出现了一批用于生物序列比对和序列特征提取的算法和系统,其中最著名的当属BLAST序列比对系统。90年代中期出现了phred-phrap-consed系统,用于鸟枪法测序中序列的碱基识别、拼装和编辑等,为人类基因组计划的实施做出了重大贡献。与此同时,一些面向生物功能预测的生物数据分析系统也开始崭露头角,如基因预测系统、RNA二级结构预测系统、核酸序列中限制性酶切位点识别系统等。进入21世纪,人类基因组计划完成,生物数据呈爆发性增长态势,生物信息系统进入了由高通量生物实验技术产生的海量生物数据的管理和分析的新的发展阶段。面向不同应用的海量生物数据库,如基因表达谱数据库、高通量测序数据库、千人基因组数据库、生物元件百科全书数据集、人类基因组单体型数据库等大量涌现;海量生物数据处理与分析系统也层出不穷,诸如序列拼接系统、片段映射系统、基因预测系统、蛋白质结构与功能预测系统、基因调控关系预测系统、药物分子设计系统等,并大量应用于生物学、医学研究的各个方面,如基因组研究、转录组研究、蛋白质组研究、表观基因组研究、遗传学研究、个体化医疗研究等重大生命科学问题研究。在后人类基因组时代,有关生物知识管理和生物数据注释的生物信息系统也开始大规模出现。其一是生物本体,包括生物医学主题词、基因本体、细胞系本体、人类表型本体、哺乳动物表型本体、解剖学本体、疾病本体等;其二是生物知识库系统,如代谢通路知识库、基因变异位点知识库、生物组织材料纳米特性知识库、基因标识知识库、面向遗传疾病的诊断系统、本体库整合系统等;其三是基于生物数据注释信息的应用系统,如各类基因组浏览器等。

根据应用目的,生物信息系统可以分为以下5类:

(1)生物数据存储和管理系统　面对海量的生物数据资源,实现生物数据的有效存储、管理和查询。生物数据存储和管理系统的典型代表是以世界三大核酸数据库为首的各类生物数据库。这些数据库存储和管理了全世界大部分的生物数据,极大地方便了数据资源共享。

(2)生物数据分析与处理系统　生物数据通常具有高通量、高噪声、数据量巨大和无序性等特点,原始数据并不能体现其蕴含的生物学意义。生物数据分析与处理系统按特定目标加工生物学数据,使数据体现出其蕴含的生物学规律,其核心是各类生物数据处理算法,涉及**算法设计**、**图论**、生物统计学、**机器学习**、**数据挖掘**等信息科学技术。这些系统主要面向重要的生命科学研究与应用,如基因组全序列获取、个体基因组分析、基因调控关系预测、蛋白质相互作用关系预测、表观遗传学机理研究等。当前,几乎每一个重要生物学问题的提出,都会伴随建立相应的生物数据分析与处理系统,用以发现生物规律,产生新的生物学知识。

(3)生物学知识库系统　生物学知识数量庞

大、结构复杂。生物学知识库系统通过建立表示模型和计算范式,使生物学知识结构化、有序化。知识表示模型的核心是生物学本体,通常嵌入在知识库中。本体对各种生物学专有名词的内涵进行了抽象和层次规划,建立了各种专有名词之间的关系,从而使各类名词的语义具备结构化的特性。生物学知识库系统的著名代表是生物通路知识库 KEGG,它是一个涵盖基因、蛋白质、代谢物等各类生物分子之间相互作用且包含各类生物通路知识的海量知识库系统。

(4) 生物学数据注释系统　对生物学数据体现的生物功能进行注释,主要包括生物元件识别和生物元件的功能注释两个方面。例如采用基因本体注释基因组上的基因和其他生物元件,人类表型本体注释疾病与表型,疾病本体注释人类基因组与疾病相关的信息等。由于这些本体注释使用的是统一的生物学词汇,消除了各系统间词汇的歧义问题。生物学家通过基因组浏览器可以人性化地标注、查找和阅读基因组数据和相应的注释信息,从而有效地了解基因组序列所包含的语义信息,进而与生物知识库系统进行关联,便能体现出生物数据内所蕴含的生物学意义。

(5) 生物学和医学应用决策支持系统　基于生物数据的存储、处理和注释,通过整合生物学知识和信息处理方法,并加入**机器学习**、**专家系统**等**人工智能**技术,对复杂的应用需求给出客观合理的辅助决策信息。

生物信息系统涉及生物学、医学、信息科学、计算机科学和软件工程等多种学科。作为交叉学科,许多挑战性课题正待人们去探索。例如,面对不断增长的海量生物数据,如何进一步降低系统核心算法的计算复杂度;为有效发现和解释复杂而多样化的生物现象,如何建立多种而不是单一或少数几种生物数据的综合处理方法;针对特定复杂生物学问题,如何提高生物信息系统计算模型的有效性,弥补因生物实验技术所限而使现有系统的数据量不足的问题等。

目前,生物信息系统的发展趋势主要表现在:①集成化　进一步从高度、广度和深度多角度整合各类生物数据和生物知识资源,使生物信息系统可以从多方面综合处理和利用生物数据发现重大生物规律。②高效化　开发更加有效的生物计算模型,使生物数据的处理在空间和时间两方面都更有效率。③个性化　开发有针对性的个体生物数据计算模型和系统,用于满足个性化医疗及其他生物和医学应用的需求。④网络化　基于先进的超级计算机、云计算及其他网络计算技术,建立适于高度并行与分布计算的网络化的生物信息系统,使系统的信息处理能力和服务水平迈上一个新的台阶。

参考文献

1. Wikipedia, Bioinfomatics, http://en.wikipedia.org/wiki/Bioinformatics

2. Galperin M Y, Fernandez-Suarez X M. The 2012 Nucleic Acids Research Database Issue and the online Molecular Biology Database Collection, Nucleic Acid Research, 2012, 40: D1

(王亚东)

shengwu xinxixue

生物信息学(bioinformatics)　研究通过信息科学方法进行生物数据的存储、检索、组织和分析的一门交叉学科。广义的定义是指应用信息科学的方法和技术,研究生物体系和生物过程中信息的存储、内涵和传递,分析生物体细胞、组织、器官的生理、病理、药理过程中的各种生物信息。狭义的定义是指应用信息科学的理论、方法和技术,管理、分析和利用生物分子数据。生物信息学研究的焦点是使用计算机数据库和计算机算法来分析蛋白质、基因和构成生物体的全部脱氧核糖核酸(基因组),获取新的生物学知识和数据分析理论与方法,推动生命科学与信息科学的双重发展。

生物信息学是 20 世纪 80 年代末随人类基因组计划的启动而兴起的分子生物学和计算机科学的交叉新学科,是建立在分子生物学的基础上的。1995 年,美国人类基因组计划第一个五年总结报告明确给出了生物信息学的定义。2001 年 2 月,人类基因组计划的完成,使生物信息学走向了一个高潮。生物信息学的发展经历了三个阶段:①基因组前期的生物信息学　主要研究序列分析、数据库的查询、计算机操作和应用;②基因组年代的生物信息学　主要研究基因的寻找、数据与数据之间的比较、网络交互界面等;③后基因组年代的生物信息学　主要研究数据的挖掘、表达、数据多样性的分析、相互交叉数据分布的总结与分析、功能基因组学等。

生物信息学的研究目标是增进人类对生物过程的理解。与其他学科相比,生物信息学的独特之处在于其着重于利用计算方法达到其研究的目的。生物信息学主要基于信息科学理论,提出新颖的计算和统计方法,建立新的数据库,来解决生物数据分析和管理中涌现的科学问题。生物信息学研究主要可

以分为两大研究领域：(1)生物数据的管理　通过发展新的工具提供各类生物数据的有效存储和获取渠道；(2)海量生物数据分析　通过发展算法和统计学方法揭示海量生物数据集中各类数据成员之间的关系。

基于这两个研究领域，生物信息学研究主要可以分为以下几个方向：

(1) 生物分子数据管理　生物分子数据量巨大，有组织地搜集和管理这些数据是各项工作的前提。

(2) 基因组全序列获取与分析　全基因组序列是基因组信息学研究的数据基础，序列获取方法主要有传统的 Sanger 测序法与新一代测序技术。针对全基因组的测序与分析主要研究内容有：①序列拼接与全基因组序列组装；②短序列比对；③不同物种全基因组序列比对；④个人基因组与群体基因组管理。随着新一代测序技术的不断进步，在不久的将来，全基因组测序将被普及，成为常规的检测手段。

(3) 数据库搜索及序列比对　其基本问题是比较两个或两个以上符号序列的相似性或不相似性。人们通过搜索序列数据库找到与新生物分子序列同源的已知序列，并根据同源性推测新序列的生物功能。

(4) 基因注释与基因组分析　分析 DNA 序列，识别基因和非编码区域，标注基因组中的基因与其他生物特征；通过分析与基因表达调控相关的信息，分析各种功能位点和基因转录调控元件等。

(5) 分子进化和比较基因组学　利用不同物种中同一基因的 DNA 序列或者氨基酸序列，通过相关蛋白质的结构比对的异同来研究生物的进化，构建进化树；建立计算模型，解释复杂的进化事件并预测未来基因组进化的结果。

(6) 基因表达分析　基因表达在分子层面决定了生物体的性状和功能。对基因表达的测量可以对通过微阵列芯片数据和高通量转录测序技术数据的分析完成。其重要工作包括微阵列数据的噪声处理、测序数据的映射，基因表达统计模型构建与统计推断等。

(7) 调控分析　基因调控是生物系统行使功能的核心要素。生物信息学研究了大量方法用于对基因间调控关系的分析。这些方法主要集中于以下两个方面：①启动子分析，通过识别基因组编码区域附近的识别和分析特性序列（如 motif 等），从序列角度探索基因转录调控的规律性特征；②基因表达数据分析，通过对全基因组表达数据的分析，挖掘基因间的相互调控关系，建立基因调控网络。

(8) 基因组多态性检测　不同生物个体的基因组含有有不同的碱基成分，从而产生了群体中个体之间的差异，这些差异包括某些单个的核苷酸的变化和较大规模的结构性改变；生物信息学研究通过对 DNA 微阵列和高通量测序数据的分析，对基因组多态性进行检测，发现这些基因组变异的情况。通过基因组多态性检测，可以发现基因组中出现的对生物表型有决定作用的变异，从而指导重大疾病的预测和治疗、生物能源开发和利用、环境净化与改良等重要生物学应用。

(9) 蛋白质结构分析　蛋白质的生物功能由蛋白质的结构所决定；通过分析蛋白质的空间结构来研究蛋白质功能，主要分为蛋白质二级结构预测和蛋白质空间结构预测。

(10) 系统与网络生物学　系统生物学主要通过生物信息学方法，进行大规模生物系统建模和数据仿真实验，对细胞内生化过程（如代谢通路、信号传导通路、基因调控）的运行原理进行研究，分析和可视化参与这些过程的生物分子交互作用。在此之上，生物信息学研究生物分子之间的网络化交互作用，以此构建整个细胞内部的各类生物分子网络（如代谢网络、蛋白质互作网络、基因调控网络等）的组成方式，拓扑结构和工作原理。

(11) 基于结构的药物设计　分析人体内约 10 万种蛋白质的结构、功能、相互作用以及与各种人类疾病之间的关系，寻求各种药物治疗和预防方法。

生物信息学在理论上促进了生物学（特别是分子生物学）的发展，使人类对生命本质的认识更加深刻。生物信息学改变了传统的生物学研究方法，提高了生物学实验的科学性和研究的效率。在应用方面，生物信息学大大促进了对人类基因组的研究，已成为生物工程、生物医药和高科技农业产业的巨大推动力，今后还将在功能基因组学研究和蛋白组学研究中发挥其突出的作用。

生物信息学是在人类基因组计划的促进下迅速发展起来的。人类对基因的认识，已从以往的对单个基因的了解，上升到在整个基因组水平上考察基因的组织结构和信息结构。随着计算机技术的发展和渗透，生物信息学在人类基因组中大规模测序的自动化控制、测序结果分析处理、序列数据的计算机管理、各类遗传图谱和物理图谱的绘制、数据的网络

获取和分析等方面发挥越来越重要的作用。生物信息学的海量数据和巨大的计算量、复杂的噪声模式、海量时变数据等给统计分析方法带来巨大挑战,需要像非参数统计、聚类分析等更加灵活的数据分析技术,并对**机器学习**算法、并行计算、**云计算**等提出了更加迫切的需求。目前,生物信息学的发展已经完全超越了人类基因组计划,向功能基因组学、蛋白质组学、系统生物学等方面发展,这使得生物信息学具有更加广阔的发展空间。

参考文献

1. 孙啸,陆祖宏,谢建明. 生物信息学基础. 北京:清华大学出版社,2005

2. Pevsner J. 生物信息学与功能基因组学. 孙之荣,等译. 北京:化学工业出版社,2006

(王亚东)

shijinzhi suanshu yunsuan

十进制算术运算(decimal arithmetic operation) 直接对两个以十进制编码的数进行算术运算。每一位十进制数可由 4 位二进制数组成,有 BCD 码和余 3 码等(参见**数制**)。

BCD 码的算术运算 用 4 位有权(8421)的二进制码来表示 1 位十进制数,例如十进制数 95 的二–十进制编码为 10010101,前 4 位表示 9,后 4 位表示 5,对两个 1 位二–十进制编码数进行加法运算时,如结果大于 9,需向高位产生进位信号,并对本位进行修正(+6);如结果不大于 9,则不需要修正。

例 1 8+9=17

$$\begin{array}{rr l} & 1000 & \\ + & 1001 & \\ \hline & 10001 & \\ + & 0110 & (+6) \\ \hline & 10111 & \end{array}$$

进位 ↗

例 2 5+6=11

$$\begin{array}{rr l} & 0101 & \\ + & 0110 & \\ \hline & 1011 & \\ + & 0110 & (+6) \\ \hline & 10001 & \end{array}$$

进位 ↗

例 3 3+5=8

$$\begin{array}{rr} & 0011 \\ + & 0101 \\ \hline & 1000 \end{array}$$

例 4 28+55=83

$$\begin{array}{rrr l} & 0010 & 1000 & \\ + & 0101 & 0101 & \\ \hline & 0111 & 1101 & \\ + & 0000 & 0110 & (\text{低位}+6) \\ \hline & 1000 & 0011 & \end{array}$$

余 3 码的算术运算 在 8421 码的基础上,每位十进制数加上 0011 形成余 3 码,其加法运算规则如下:

当两个余 3 码相加不产生进位时,应从结果中减去 0011;产生进位时,应将进位信号送到高位,本位加 0011。

例 5 28+55=83

$$\begin{array}{rrrr l} & 0101 & & 1011 & \\ + & 1000 & & 1000 & \text{低位向高位产生进位,} \\ \hline & 1110 & & 0011 & \text{高位不产生进位,低位}+3, \\ - & 0011 & + & 0011 & \text{高位}-3 \\ \hline & 1011 & & 0110 & \end{array}$$

(王爱英)

shiduan yansuan

时段演算(duration calculus) 一种实时区间**时态逻辑**。它将布尔函数在区间上的积分进行形式化,从而用来描述和推导离散状态系统的实时和逻辑特性。

时段演算的研究始于 1989 年。当时 Esprit 的研究项目 ProCoS 正寻求设计严格安全系统的形式技术,应用项目的需要推动了时段演算的研究。该演算是由周巢尘、C. A. R. Hoare 和 A. P. Raun 所提出。

时段演算已应用于若干实例,如煤气燃烧器、铁路岔口控制、水位控制、自动导航、Occam 语言的实时语义、描述调度程序的实时行为和电路设计等方面。

ProCoS 项目的一个研究实例就是要对如下煤气燃烧器需求进行形式化:“如果对系统观察的时间大于 60 s 时,那么漏气的时间占整个时间的比例应小于 1/20。”应用数学分析可直接地对这个需求进行形式化,其结果是

$$(e-b) \geqslant 60\ \mathrm{s} \Rightarrow 20\int_b^e \mathrm{Leak}(t)\,\mathrm{d}t \leqslant (e-b)$$

这里 Leak 是一个布尔函数，它表示煤气燃烧漏气状态。函数是从实数 R(表示时间)到{0,1}的函数，其中1表示系统正处于该状态，0表示系统不处于该状态。观察的区间采用闭区间，并且用 b 表示开始和用 e 表示结束。积分被认为是从状态函数和区间到实数的函数：$\int: S\rightarrow(I\rightarrow R)$。其中 S 表示状态集(即布尔函数)，I 表示闭区间集。因此区间时态逻辑(这里被扩展成连续时间模型)被作为它的基础逻辑，并且区间函数 $\int S$, $\int P\cdots$变成了演算的区间变量。这里 S 和 P 是状态。如此有 $\int l=e-b$，并且我们用 l 作为它的缩写，即是区间的长度。因此上面的需求可以更简单地描述为：

$$\text{Req: } l\geqslant 60\Rightarrow 20\int \text{Leak}\leqslant l$$

通过用积分定义一个$\lceil\cdot\rceil$的运算，我们可以表示一个状态在区间上持续的出现：$\lceil S\rceil\triangleq(\int S=l\wedge l>0)$，$\lceil S\rceil$在一个区间上成立要求这个区间为非点区间，并且状态 S 在区间上(几乎)处处取值为1。如此下面的公式

Dec1：$(\lceil\text{Leak}\rceil\Rightarrow l\leqslant 1)$

形成了煤气燃烧器可行的设计决策，即一次失败的点火在1 s内是可以被检测到而且能够被停止的。

区间时态逻辑的模态词是“切变”(;)，语义上定义为

$A;B[b,e]\triangleq\exists m:\ b\leqslant m\leqslant e.\ A[b,m]\wedge B[m,e]$

它的含义是公式 $A;B$ 在区间$[b,e]$上成立，当且仅当区间可以被切成两个子区间，并且使得 A 在第一个区间上成立，B 在第二个区间上成立。切变是个连接运算，它使设计者能够通过组装几个子区间上的性质，从而刻画整个系统在观察区间上的性质。例如利用切变，可以对煤气燃烧器另外一个设计决策进行形式化。它要求系统在两次漏气之间至少要等待30秒，从而使我们排除了煤气燃烧器频繁的点火故障。

Dec2：$(\lceil\text{Leak}\rceil;\lceil\neg\text{Leak}\rceil;\lceil\text{Leak}\rceil\Rightarrow l\geqslant 30)$

通常的模态词可以由切变来定义：$\Diamond A\triangleq\text{true};A;\text{true}$ 和 $\Box A\triangleq\neg\Diamond\neg A$。$\Diamond A$ 在一个区间上为真是指 A 在某一个区间上为真；$\Box A$ 在一个区间上为真是指 A 在每一个区间上都为真。

假设状态具有有限变化性。我们可以得到6个公理。它们类似测度论中关于有限区间集合的测度公理，并且构成了一个相对完全的演算，即时段演算。它们是

1. $\int 0=0$
2. $\int P\geqslant 0$
3. $\int P+\int Q=\int(P\vee Q)+\int(P\wedge Q)$
4. $\left(\int P=r+s\right)\Leftrightarrow\left(\int P=r\right);\left(\int P=s\right)$

$(r,s\geqslant 0)$

5. $(\lceil\rceil\vee\lceil P\rceil;\text{true}\vee\lceil\neg P\rceil;\text{true})$
6. $(\lceil\rceil\vee\text{true};\lceil P\rceil\vee\text{true};\lceil\neg P\rceil)$

其中$\lceil\rceil\triangleq(l=0)$。公理5和6刻画了状态 P 的有限变化性。也就是，对于任何非点区间都可以分成有限次 P 和$\neg P$ 的交换。

应用该演算，可证明下面的公式是该演算的定理，从而证明了上面两个设计决策的确满足需求的正确实现。

$\Box\text{Dec1}\wedge\Box\text{Dec2}\Rightarrow\text{Req}.$

在实时系统形式化领域的研究中，时段演算被公认是一种成功的形式化方法。目前时段演算的研究还在不断地发展完善之中。

参考文献

1. Zhou Chaochen. A calculus of durations. Information Processing Letters, 1991, 40(5)

2. 李晓山，周巢尘. 时段演算综述. 计算机学报，1994，17(11)，842-851

(李晓山　周巢尘)

shifen fuyong

时分复用(time division multiplexing, TDM)

采用分时技术把传输线路的可用时间分成时隙，并按一定规律将时隙分配给多个输入信号的复用技术。

时分复用系统中，通常传输时间被分成若干周期，如图中的T1、T2、T3等，每个周期又被分为若干时隙，每个时隙被分给各终端用户，各终端用户在自己的时隙才能发送数据；在每个周期，各用户轮流按约定时隙发送数据，构成的数据流叫TDM帧，在复用线路上传输。时分复用具体工作过程为：

(1) 接到复用器的每个终端都分配到一个时隙；

(2) 复用器对每个终端装置进行扫描，以确定它是否有字符要传送，如果有，将其放到复用的线路上；

(3) 如果终端装置在自己的时隙没有字符要传送,复用器便发送一个空(零)字符,以保持序列顺序;

(4)复用线路上的信号到达接收端复用器,复用器按照约定时隙,顺序读取,可分离出各终端发送的原始信号。

如图1所示,各终端的数据传输速率之和,即为复用线路的数据传输速率。图1中的3个终端可以推广到 N 个终端,即复用的信号路数可以是2个或以上,只受制于复用器和线路的能力。

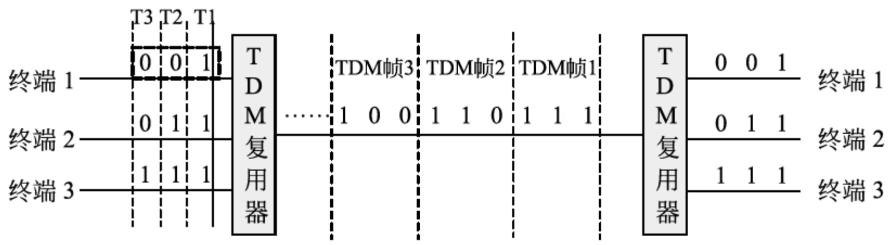


图1　时分复用原理示意图

需要注意的是,时分复用技术通常适用于数字信号,很少用于模拟信号;每个时隙,各用户可发送1个比特、字符、字节或多个比特。

时分复用技术是一种传统的经典的复用技术,用于公共电话网(PSTN)、全球移动体内更新系统(GSM)中。下面以公共电话网络为例,介绍TDM技术的应用实例。

公共电话网络的最后一千米(本地回路,local loop)传输的是模拟信号,在端局汇聚,并需要将模拟信号转换为数字信号,采用PCM(pulse code modulation)技术来完成这个转换。由于CCITT未能就PCM技术达成一致的国际标准,导致互不兼容的两种方案,这就是我们熟知的T1线路(北美和日本使用)和E1线路。T1线路包含24条复合在一起的语音信道,每秒采样8000次,时隙是125 μs,每个时隙产生193位数据,其中的控制信令有25位;T1线路的传输速率是193bit/125 μs = 1.544 Mbps。采用时分复用技术,可以将T1线路复用到更高级别的线路上,如图2所示。

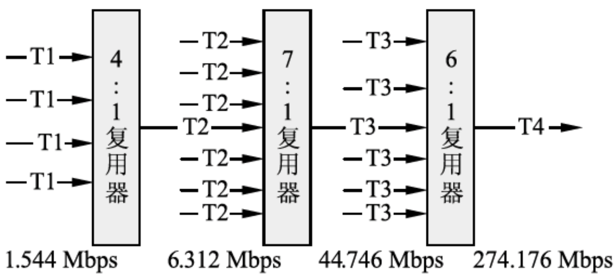


图2　更高级别的T1时分复用示意图

时分复用技术比较适用于每个终端用户都有数据发送的情况,如果某个终端发送的数据是稀疏的,将会导致分配给它的时隙无数据发送,造成信道的浪费,这种时分复用又称为同步时分复用。为了避免空闲时隙的产生,有效利用带宽,又出现了异步时分复用,也叫统计时分复用,它可以动态地按需分配时隙,避免空闲时隙的产生。

参考文献

Tanenbaum A S, Wetherall D J. 计算机网络. 北京: 清华大学出版社, 2012

(袁华)

shijian fuzaxing

时间复杂性(time complexity)　用时间来度量的计算复杂性,它是最重要的**复杂性度量**。粗略地说,算法的时间复杂性就是该算法的执行时间。算法 C 对每一个可能的输入长度 x,若执行时间为 $t(x,C)$,则 $W_t(n,C) = \max\{t(x,C) \mid x$ 的长度为 n,即 $|x| = n\}$ 给出用该算法解输入规模为 n 的问题所需要的最长时间。称这种时间复杂性为最坏情况时间复杂性,如果把 $D_n = \left\{x \;\middle|\; |x| = n\right\}$ 看成一个概率空间,出现输入 x 的概率等于 $p(x)$,于是算法解输入规模为 n 的问题所需要的平均计算时间为 $A_t(n,C) = \sum_{x \in D_n} p(x)t(n,C)$。计算时间与执行算法的计算机或计算模型有很大的关系。如在图灵机模型中,计算时间是指图灵机停机前执行的总步骤数。但从理论上讲,根据相似性和对偶性原理:同一算法在各种计算模型上计算时间复杂性并无很大的差别。在实用上,常用基本操作数作为时间的度量,例如对排序问题,以被排序元素的比较次数为基本操作数,又如矩阵乘法以矩阵元素的相乘次数为基本操作度量,这样计算时间的目的是把影响算法

执行时间的次要因素舍弃，而着重计算影响算法执行时间的主要因素——基本操作数，从而简化了时间复杂性的计算但又不损害对该算法性态的评价。

时间复杂性函数在 $\bigcup_k O(n^k)$ 中的算法称为多项式时间算法，超过此范围的，统称为指数时间算法。对于下面5个算法：

算法	时间复杂性
A_1	n
A_2	$n\log n$
A_3	n^2
A_4	n^3
A_5	2^n

表1、表2分别给出特定时间内它们能解决问题的规模和计算机速度增加10倍后它们计算能力的增强情况。

表1　复杂性限制的问题大小

算法	时间复杂性	最大问题规模		
		1s	1min	1h
A_1	n	1000	6×10^4	3.6×10^6
A_2	$n\log n$	140	4893	2.0×10^5
A_3	n^2	31	244	1897
A_4	n^3	10	39	153
A_5	2^n	9	15	21

表2　计算机速度提高10倍的效果

算法	时间复杂性	加速前最大问题规模	加速后最大问题规模
A_1	n	s_1	$10s_1$
A_2	$n\log n$	s_2	$4.6s_2$
A_3	n^2	s_3	$3.16s_3$
A_4	n^3	s_4	$2.15s_4$
A_5	2^n	s_5	$s_5+3.3$

从上表可以看出，对多项式时间算法，计算机计算能力的提高对问题的可解规模明显增加，而对指数时间算法，计算机计算能力的提高使问题的可解规模几乎不增加。因此计算机科学家公认简单区分算法效率高低的分界线是它为多项式时间算法还是指数时间算法，凡一个问题有多项式时间算法则称该问题是易解的，称它属于P类。反之，如一个问题还没有多项式时间算法则称该问题是难解的。寻找问题的多项式时间算法或者证明该问题是难解的一直是算法研究的最重要的内容。如果问题有不确定的多项式时间算法，则称它属于NP类。NP类中最难解的问题称为NP C类。P类是否就等于NP类是当代理论计算机科学中最著名而重要的未决难题。

随着计算机系统向并行、并发、分布式和巨型体系发展，遂有并行、分布式算法的兴起，并行算法的时间复杂性和空间复杂性有相互折算的关系，空间耗费大，即并行程度高，则并行算法的执行时间可以降低。反之，算法的并行程度低，则并行算法的执行时间就会增加。设计并行时间复杂性态好的并行算法和分析并行算法的**时空折算关系**以及一个问题的并行算法时间复杂度下界是并行算法研究主要关心的问题。

参考文献

1. 朱洪，段振华，陈增武，周克成. 算法设计和分析. 上海：上海科学技术文献出版社，1989

2. Cormen T H, Leiserson C E, Rivest R L. Introduction to Algorithms. Cambridge, MA: The MIT Press, 1990

3. Akl S G. The design and analysis of parallel algorithms. NJ: Prentice Hall, 1989　　（朱洪）

shikong shuju wajue

时空数据挖掘（spatio-temporal data mining）

对时空数据的挖掘（参见**数据挖掘**）。旨在从大规模时空数据中获取有效、潜在有用和可理解的知识，及实现这一目标所涉及的理论、技术与方法的研究。

早期的数据挖掘研究主要针对字符-数值型的商业数据。20世纪90年代中后期，数据挖掘领域的一些较成熟的技术，如关联规则挖掘、分类、预测与聚类等被逐渐用于时间序列数据挖掘和空间数据挖掘，以发现与时间或空间相关的有价值的模式。随着**传感网**、**全球定位系统**（GPS）、手持移动设备和射频识别（RFID）等技术的普遍应用，积累了大量时空数据（同时具有时间和空间属性）。这些时空数据呈现出内嵌于连续空间，其样本在时间、空间上存在很强的自相关性，其中隐含的模式往往是局部的等特点，从而使时空数据挖掘具有特殊性和复杂性。寻找针对时空数据的挖掘技术和方法成为新的研究课题。2005年以来，时空数据挖掘逐渐成为一个独立的研究领域，并已在交通控制、灾害预测、环境保护、公共卫生保健和移动电子商务等领域得到较广泛的应用。

按挖掘任务来分，时空数据挖掘技术主要包括：

时空模式发现　时空模式主要包括时空频繁模

式(spatiotemporal frequent pattern)、时空互现模式(spatiotemporal co-location pattern)、时空周期模式(spatiotemporal periodic pattern)等。时空互现模式是指两种或两种以上对象在空间和时间上处于近邻;时空周期模式是指对象在固定时间间隔内遵循相同或近似相同的移动路线,显示出一定的周期性规律。时空模式发现在优化资源配置、动物迁移模式研究、天气预报等方面具有重要的应用价值。

时空关联分析(spatiotemporal association rule) 主要研究空间对象随时间发生变化的规律,即在传统关联分析的基础上加上了时间和空间约束,以发现时空数据中处于一定时间间隔和空间位置的关联规则。时空关联分析具有重要的应用价值,如研究战场上的战术、调查动物捕食关系等。

时空聚类(spatiotemporal clustering) 是指基于空间和时间相似度把具有相似行为的时空对象划分到同一组中,使组间差别尽量大,而组内差别尽量小。时空聚类可用于交通拥挤预测、动物迁移分析、移动计算和异常分析等方面。

时空异常检测(spatiotemporal outlier detection) 时空异常对象是指一个和它在空间上相邻并在一段连续时期内出现的邻居有着显著差异的对象。时空异常检测旨在时空数据中找出严重偏离正常模式的对象。目前已被广泛应用于健康和医疗监控、可疑移动对象检测等方面。

时空分类和预测(spatiotemporal classification and prediction) 基于时空对象的特征构建分类模型来预测时空对象所属类别或对象所在具体空间位置。对于实时物流、实时交通管理、基于位置的服务和 GPS 导航等涉及时空数据的应用而言,预测单个或一组对象未来的位置或目的地是至关重要的。

时空数据挖掘研究需要汇集**数字图像处理**、**模式识别**、**地理信息系统**、并行处理和统计数据分析等多学科知识。目前,图像和视频的自动分类、时空数据分类、频繁/连续模式和异常值发现、空间配置分析等研究已取得较大进展。然而,时空数据挖掘研究仍面临诸多关键难题,包括:如何抽取数据中隐含的复杂时空关系,如何避免由此引入的模糊性和不确定性对挖掘结果的影响;如何给出有效的复杂转换方法,以便在更高概念层面上对数据进行描述,发现易于理解的模式;如何在多种粒度/分辨率层面上进行时空数据的挖掘;如何表达领域独立的知识;如何在数据挖掘系统中(深度)结合时空推理等。

参考文献

1. Kargupta H, Han J, Yu P S, et al. Next generation of data mining. CRC Press, Taylor & Francis Group, 2009

2. Miller H J, Han J. Geographic data mining and knowledge discovery. 2nd ed. CRC Press, Taylor & Francis Group, 2009

(刘大有 齐红)

shitai luoji

时态逻辑(temporal logic) 关于随着时间变化而不断改变其值的动态变元(称为时序变元)的一种**模态逻辑**。它除含有经典逻辑的逻辑联结词和量词外,还含有一些时态算子。

时态逻辑包括命题时态逻辑和谓词时态逻辑,选择不同的时态算子将导致不同的时态逻辑系统。在计算机科学中应用较为广泛的命题线性时态逻辑系统 PLTL,是由 A. Pnueli 和 Z. Manna 给出的。PLTL 包含可数无穷多个命题变元,以及逻辑联结词 $\neg$(否定)、$\wedge$(合取)、$\vee$(析取)、$\supset$(蕴含)与$\equiv$(等价)和时态算子$\Box$(意为“任一时刻”)、$\Diamond$(意为“某一时刻”)、$\bigcirc$(意为“下一时刻”)与$\mathcal{U}$(意为“直到”)。PLTL 的合式公式可归纳定义如下:

(1) 命题变元 P 是合式公式;

(2) 若 w, w_1 和 w_2 是合式公式,则($\neg w$)、$(w_1 \wedge w_2)$,$(w_1 \vee w_2)$,$(w_1 \supset w_2)$和$(w_1 \equiv w_2)$均为合式公式;

(3) 若 w, w_1 和 w_2 是合式公式,则$(\Box w)$,$(\Diamond w)$,$(\bigcirc w)$和$(w_1 \mathcal{U} w_2)$均为合式公式;

(4) 每个合式公式皆可通过有限次应用(1),(2),(3)得到。

PLTL 包含以下 10 条公理和 3 条推理规则:

公理 1: $\neg\Diamond w \equiv \Box \neg w$

公理 2: $\Box(w_1 \supset w_2) \supset (\Box w_1 \supset \Box w_2)$

公理 3: $\Box w \supset w$

公理 4: $\bigcirc \neg w \equiv \neg \bigcirc w$

公理 5: $\bigcirc(w_1 \supset w_2) \supset (\bigcirc w_1 \supset \bigcirc w_2)$

公理 6: $\Box w \supset \bigcirc w$

公理 7: $\Box w \supset \bigcirc \Box w$

公理 8: $\Box(w \supset \bigcirc w) \supset (w \supset \Box w)$

公理 9: $(w_1 \mathcal{U} w_2) \equiv (w_2 \vee (w_1 \wedge \bigcirc(w_1 \mathcal{U} w_2)))$

公理 10: $(w_1 \mathcal{U} w_2) \supset \Diamond w_2$

命题重言规则:若 u 是命题重言式,则 $\vdash u$

假言推理规则:若 $\vdash u \supset v$ 且 $\vdash u$,则 $\vdash v$

$\Box$引入规则:若 $\vdash u$,则 $\vdash \Box u$

应用上述公理和推理规则经有穷步可推导出来的合式公式称为该系统的定理。

在时态逻辑中,时间的结构可以是线性的或分支的,离散的或连续的,基于时间点的或时区的。基于不同的应用背景,可采用不同的时间结构。PLTL采用线性的、离散的且与自然数同构的时间结构。PLTL的语义解释是一个无穷状态序列 $\sigma = s_0, s_1, s_2, \cdots$,其中每个 s_i 是对命题变元的一个赋值。若令 $\sigma^{(i)} = s_i, s_{i+1}, \cdots$,且用 $\sigma \vDash w$ 表示时态公式 w 在解释 σ 下为真,则时态算子的含义定义如下:

$\sigma \vDash \Box w$　当且仅当　对任意 $i \geqslant 0$,均有 $\sigma^{(i)} \vDash w$

$\sigma \vDash \Diamond w$　当且仅当　存在 $i \geqslant 0$,使 $\sigma^{(i)} \vDash w$

$\sigma \vDash \bigcirc w$　当且仅当　$\sigma^{(1)} \vDash w$

$\sigma \vDash w_1 \mathscr{U} w_2$　当且仅当　存在 $i \geqslant 0$,使 $\sigma^{(i)} \vDash w_2$ 且对任意 $j(0 \leqslant j < i)$ 均有 $\sigma^{(j)} \vDash w_1$

时态逻辑的发展是与程序规约和程序验证紧密相关的。程序的行为是一种动态现象,其状态随着时间的推移不断改变,且可能不断影响外部环境。持续不终止的并发反应式程序的动态行为是经典逻辑和**霍尔逻辑**所不能描述的。为此,R. Burstall 在1974年首先建议使用模态逻辑进行程序推理,A. Pnueli和 Z. Manna 建立了可用于程序规约和验证的时态逻辑系统。

时态逻辑具有很强的表达能力,可表达程序的安全性(如部分正确性、互斥性和无死锁性等)、活性(如终止性、完全正确性和响应性等)和事件优先性,是一种研究并发程序尤其是持续不终止的反应式程序(如操作系统、网络通信协议等)的强有力的形式化工具。目前,时态逻辑已广泛应用于程序的规约、验证和形式化开发,以及程序自动综合和模块化规范合成等并发程序设计的几乎所有方面。通过引入全局时钟或在时态算子上增加时间约束,时态逻辑亦可用于实时系统的规约和验证。

参考文献

Emerson E A. Temporal and modal logic. In: Handbook of Theoretical Computer Science. Elsevier Science Publishers, 1990

(李舟军)

shitai shujuku

时态数据库(temporal database,TDB)　数据中包含了时间属性的数据库。时态数据库不但存储数据本身,还存储与数据有关的时间,并能有效地管理它们。带有时间属性的数据被称为时态数据,相应的属性被称为时态信息。传统数据库没有对时态数据做专门的处理,只反映现实世界的当前状态,不反映历史状态。管理被处理事件的历史性信息(如与自然灾害有关的历史资料)或数据库中元事件的时态信息(如更新事务的时间)的应用需求催生了时态数据库,对时态数据库的研究始于20世纪70年代,其应用领域涉及**地理信息系统**、电信信息系统、电子政务、电子商务、数据仓库以及**决策支持系统**等。

时态数据库的时间属性一般包括用户自定义时间、有效时间和事务时间。用户自定义时间指用户根据自己的需要或理解定义的时间;**有效时间**(valid time)是指一个数据对象在现实世界中发生并保持的那段时间,或者该对象在现实世界中为真的时间,有效时间可以反映过去、现在或将来的时间;**事务时间**(transaction time)是指对数据库对象进行操作的时间,记录对数据库修改或更新的各种操作历史,对应于现有事务或现有数据库状态变迁的历史,例如,录入、查询、修改、删除数据对象的时间。

按表示时态信息的方式可以将时态数据库分为回滚数据库、历史数据库和双时态数据库三个基本类型。**回滚数据库**(rollback database)支持事务时间,按事务时间进行编址,它保存了每个事务提交及事务执行状态演变之前的状态;**历史数据库**(historical database)支持有效时间,历史数据库记录了在有效时间轴上的一系列数据库状态,可能是过去、现在或将来,它记录了事实在真实世界中的变化过程;**双时态数据库**(bitemporal database)既支持事务时间又支持有效时间,是两者的结合。

时态数据库管理系统对传统的关系数据库管理系统进行扩充来处理增加的时间属性。时态关系代数将基本关系代数运算推广到时间域,以支持离散线性有界时间的时态关系数据模型。为提高系统对时态数据的支持能力和减少系统的开销,需要按时间建立索引,以实现时态聚集函数处理、时态选择和时态连接等操作。因为数据在时间坐标上展开后,数据量将大幅增加,同时系统要将时态数据作为复杂对象来管理和处理,具有一定的复杂性和难度。

时态数据库管理系统的典型代表是 Andreas Steiner 等人开发的 TimeDB,它支持建立在 SQL 上的时态查询语言和时态完整性约束,是建立在传统 DBMS 之上的前端软件。它接收时态查询语言 TSQL2 描述的数据库查询语句,并将它们转换成 SQL92,实现基本的时态数据管理功能。

参考文献

1. Tansel A, Clifford J, Gadia S, et al. Temporal database: theory design and implementation. The Benjamin Cummings Publishing Company, 1993

2. 李昭原,等. 数据库技术新进展. 2 版. 北京:清华大学出版社,2007

(周龙骧 王翰虎)

shixu xitong

时序系统(timing system) 产生控制计算机各部件工作的控制命令的定时信号,指定有关控制命令在**指令周期**中的时间关系。时序系统包括时钟脉冲与时钟周期、节拍电位与工作脉冲、机器周期、指令周期等。

计算机通过执行具有确定算法的指令序列,完成对信息的处理和加工。如何指挥系统中各种独立的功能部件,协同完成程序和指令规定的操作,保证系统高速运行,这项艰难的使命是由**控制器**完成的。控制器是计算机的指挥中心,是计算机的大脑,时序系统是计算机的心脏。

时序系统向计算机各部件提供定时信号,使有关部件在规定的时间内完成规定的操作。完成一条机器指令的时间叫一个指令周期,一个指令周期又分成若干个机器周期(又叫 CPU 周期),每个机器周期又分成若干个节拍,执行若干个微操作,每个节拍又设置若干工作脉冲,形成一个严密的时序系统(图 1)。

时序系统举例:如果所执行的指令有两个机器周期:取指周期和执行周期,每个周期设置四个节拍,每个节拍设置一个工作脉冲,则其时序信号波形如图 1 所示。第一个机器周期 M_1 是取指周期:第一节拍 T_1,通过地址总线发送指令地址,其工作脉冲 m_1 把指令地址打入内存地址寄存器。第二节拍 T_2 经过控制总线发读内存命令。第三节拍 T_3 监测内存读出操作是否完成,等待内存读出。如果内存读出完成,发出准备好信号 READY,转入第四节拍 T_4,内存读出的指令,经数据总线送到控制器,其工作脉冲 m_4 把指令打入**指令寄存器**中。如果 T_3 节拍未能读出要求的内容,内存没有发出 READY 准备好信号,则在 T_3 与 T_4 节拍之间插入一个等待节拍 T_w,在等待节拍中继续监测准备好信号,若在 T_w 等待节拍采样到准备好信号 READY,本节拍结束后进入 T_4 节拍,读入指令,否则继续插入 T_w 节拍继续等

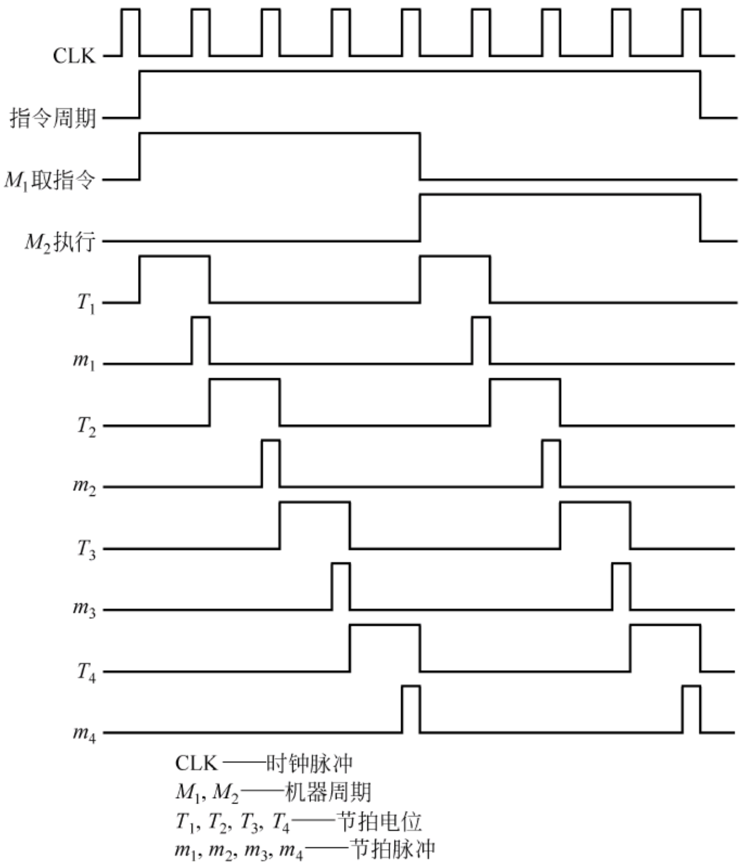


图 1 时序信号波形图

待，直到读出完成。

第二个机器周期 M_2 是执行周期，完成不同指令规定的不同功能。

需要注意，由于指令功能的差异，不同指令需要的机器周期的数目是不同的，指令周期也是不同的。

时钟周期是 CPU 操作的基本时间计量单位，是由计算机的主频决定的。

参考文献

1. 金兰，金波. 计算机组织：原理、分析与设计. 北京：清华大学出版社，2006

2. 谢树煜. 计算机组成原理. 2 版. 北京：清华大学出版社，2009

（谢树煜）

shizhong fashengqi

时钟发生器（clock generator）　同步时序电路产生时钟信号的器件或装置。最简单的时钟发生器是由奇数个反相器串接而成的环形振荡器，通过控制反相器的个数和/或连线的长度（延迟），可以得到不同周期的方波信号。由于器件参数的离散性，这种环形振荡器产生的时钟信号精度有限，一般只用于频率较低的简单系统中。

更常见的做法是用性能稳定的压电石英晶体振荡器产生固定频率的谐振信号，再根据实际使用的需要，配以附加电路，生成满足要求的时钟信号。这类附加电路的功能主要包括：①改变周期信号的占空比，如从 1∶1 变成 2∶3；②生成多相时钟；③时钟分频；④时钟倍频；等等。

高频数字系统对时钟信号的精确性和稳定性有严格的要求。“抖动”是衡量时钟信号质量的一个重要指标，它体现了信号在幅度、相位等方面与理想状况的偏差。导致抖动的因素很多，如电磁干扰、传输反射、线间串扰等。为了尽可能缩小抖动及它带来的不利影响，需要采用专门的附加电路结构，如锁相环和延迟锁定环。

锁相环，顾名思义，是用来锁定时钟的相位、防止相位偏离理想状态的。它是一个反馈式的模拟电路，根据输入时钟与输出时钟的相位差，调整压控振荡器的输出，使输出时钟与输入时钟的相位保持一致。通常锁相环可输出多个不同周期的时钟信号，还带有倍频和分频的功能。延迟锁定环的功能与锁相环类似，它根据输入时钟与输出时钟的相位差，调整压控延迟线的输出，使输出时钟与输入时钟的相位保持一致。延迟锁定环可由数字电路实现。

实际的计算机系统因为不同的需要，在不同的场合可能会工作在不同的时钟频率上。例如，同一块主板上可以安装标称频率各不相同的多款 CPU 芯片。为了满足这样的要求，时钟发生器应具备一定的可配置能力，如通过主板上的跳线或开关，设定不同的输出频率。更进一步的做法是由主板上的基本输入输出系统（BIOS）或固件来设定时钟发生器的输出频率，称为可编程的时钟发生器。一些电脑爱好者在玩超频游戏时对 CPU 主频的修改，就是通过设置可编程时钟发生器来完成的。

近年来，为了尽可能地降低计算机的功耗，许多处理器开始采用动态频率调整技术，即在系统负荷较轻时，把处理器主频调整到较低的水平。这种技术对时钟发生器也提出了新的要求，即要求时钟发生器输出的时钟频率能够在不影响计算机高速运行的前提下实时地调整，或升高，或降低。　（唐志敏）

shishi caozuo xitong

实时操作系统（real-time operating system，RTOS）　能够在指定或确定的时间限制内完成应用任务执行及其输入/输出操作要求的操作系统。

实时操作系统不仅具有在事先定义的时间内识别和处理离散事件的能力，而且具有在规定时间内运行和管理应用任务的能力。与一般通用操作系统相比，实时操作系统的不同之处在于它具有以下特点：

（1）时间约束性　系统所管理的周期任务、偶发任务、非周期任务等不同类型的实时应用任务（进程）均要求满足时间限制，因此，实时操作系统的任务调度技术，无论抢占式优先级或截止期优先等调度策略，还是时钟同步机制，均以尽可能地保证每个任务满足其时间约束为目标。

（2）可预测性　指操作系统能够对实时任务的执行时间进行判断，确定是否能够满足任务的时限要求。其涉及硬件平台延迟的可计算性、操作系统实时原语执行的有界性以及多应用任务执行时间的可预测性等。为此，实时操作系统时常需要对关键任务采取资源预留策略，需要设计容错与恢复以及保证在系统异常情况下，关键任务的顺利执行。

（3）与外部环境的交互作用性　系统通常处在与物理世界交互的运行环境之中，它必须在规定的时间内对外部环境变化所发出的处理请求及时做出响应。由于外部环境的动态变化、外部事件的并发出现，使得实时操作系统必须具有高效有序的并发事件识别、管理与处理能力。

目前主流的实时操作系统有 VxWorks、RTLinux、QNX、LynxOS、μC/OS-Ⅲ 等，这些具体的实时操作系统虽然功能与性能有所差别，但均不同程度的体现

了以上实时操作系统的特点。

随着计算技术及其应用的日趋深化和广泛，实时系统的多核化硬件平台、网络化运行环境和智能化应用处理特征也更加鲜明，这将进一步增加系统状态与行为的可预测性和开放环境下分布实时任务调度等技术难度。

参考文献

1. Penumuchu C V. Simple real-time operating system: a kernel inside view for a beginner. Trafford Publishing, 2007

2. Silberschatz A, Galvin P B, Gagne G. 操作系统概念. 7 版. 郑扣根，译. 北京：高等教育出版社，2010 （周兴社 张羽）

shishi chuli

实时处理（real-time processing） 对数据或信息进行的一种快速而及时的处理方式。实时处理所花费的时间具有严格约束，处理结果可以立即用于响应或控制被监测的对象或过程。实时数据或信息处理的正确性不仅依赖于系统计算的逻辑结果，还依赖于产生这个结果的时间。

实时是对时间的定量描述，通常使用物理时钟进行测量；响应时间是实时处理系统的重要指标，其是从某一事件发生到系统对该事件有所响应所需的时间。

实时处理的时间约束与处理系统服务的具体对象和物理过程紧密相关，依据应用需求与特性，可将实时处理分为“硬实时”和“软实时”两类。

“硬实时”指应用任务必须在规定的时间期限内完成处理，否则被控系统将导致严重后果。在航空航天、军事、核工业等一些关键领域中，具有鲜明的“硬”实时需求与特性，若任务处理超越时间约束，可能造成如飞机失事等重大安全事故，造成重大生命财产损失和生态破坏。

“软实时”意为某些应用任务虽然提出了时间约束需求，但实时处理偶尔违反这种需求，对系统的运行以及环境不会造成严重影响。如视频点播、信息采集等应用是典型的软实时实例。在视频点播应用中，系统只需保证绝大多数情况下视频数据能够及时传输给用户，偶尔的数据传输延迟只会产生性能下降，对用户不会造成大的影响。

在实时处理过程中，被处理的信息进入系统的时间带有一定的随机性，因而难免出现任务超过系统处理能力的短暂过载，实时处理技术必须考虑在保障关键任务时间约束下的过载预防或处理机制；实时处理要求系统具有很高的可靠性，信息的任何错误或丢失均可能造成严重的损失，因此，必须采用相应的硬件冗余和软件例外处理等措施，提高系统的可靠性。

为了有效实现实时处理，计算系统的硬件平台、实时操作系统、实时数据库、实时通信以及实时数字图像处理等不同层面不仅需要协同操作，而且应采用支撑实时处理的新技术与新机制。

在后 PC 时代，快速 FPGA、多核与众核等硬件技术的发展，为实时处理提供了更为良好的物质基础；物联网的快速发展与深化应用，实时处理技术呈现出鲜明的网络化、智能化新特点。实时处理技术既有新的发展机遇，又有新的技术挑战。

参考文献

1. Liu J W S. Real-time systems. Prentice Hall, 2000

2. Krishna C M, Shin K G. 实时系统. 戴琼海，译. 北京：清华大学出版社，2004 （周兴社 张羽）

shishi huizhi

实时绘制（real time rendering） 一类可以在任意视点条件下，至多在 1/24 s 内完成整个场景的绘制并输出一帧图像的过程和技术。在 1/24 s 内必须完成绘制过程并输出结果，将保证在用户看来这些生成的图像序列是连续变化的。实时绘制的要求是要在保证时间约束的前提下，尽可能提高所绘制图形的真实感。实时绘制技术主要用于交互式三维设计、三维计算机游戏、实时仿真器、虚拟现实系统等场合。

一般来说，场景是由一系列具有不同材质的造型元素组成的。普通的绘制算法的效率与景物空间复杂度相关，即生成一幅场景画面，需要遍历并处理场景中的每一个造型元素，至少具有 $O(N)$ 的计算复杂度（N 为场景中造型元素的个数）。然而，理论上，场景的复杂性是无限的，且任何硬件在单位时间内的处理能力总是有限的，所以具有 $O(N)$ 算法复杂度的绘制算法难以胜任实时绘制的任务。因此，实时绘制方面研究的关键在于通过研究场景复杂性和人类视觉感知规律等相关因素的内部联系，设计出与场景复杂性基本不相关或弱相关的绘制算法，即具有 $O(1)$ 或 $O(\log(N))$ 算法复杂度的绘制算法。

常见的实时绘制技术包括可见性剔除、层次细节、**基于图像的绘制**等。

（1）可见性剔除 这类算法可以用较小的代价判断哪些造型元素一定不可见。当场景被组织成层次结构时，对任意观察参数，可见性剔除算法在绘制

过程的较早阶段以较少的处理时间成批地淘汰掉不可见的造型元素，这样可使得绘制时间不随场景的规模而线性增长。

（2）层次细节　其基本原理是，利用透视投影的特性——距离当前观察视点越远的物体，在成像平面上的投影面积越小，那么远处的物体在绘制阶段可用较少的等效造型元素来表现它，这使得绘制时间大大缩短。

（3）基于图像/视频的绘制方法　该方法首先在一系列视点上对整个场景进行预绘制，并将获得的图像经适当组织形成图像库。然后在绘制过程中，根据任意观察参数，从图像库中抽取出合适的图像像素或块，再经过适当处理后，来填充待绘制图像上的像素或区域。

（4）预计算辐射传输方法　该方法针对静态场景，通过对场景中辐射传输的预计算分解与投影，用一组基函数进行逼近。在绘制时，利用基函数间的正交性，可以大大加快绘制方程中辐射传输与光照的积分的计算，实现对静态场景的实时重光照。

参考文献

1. Akenine-Moller T, Haines E. Real-time rendering. 2nd ed. A. K. Peters Ltd., 2002

2. Cohen-Or D, Chrysanthou Y, Silva C. A survey of visibility for walkthrough applications. In: Proc of EUROGRAPHICS'00, Course Notes, 2000

3. Heckbert P S, Garland M. Survey of polygonal sur-face simplification algorithms. Course SIGGRAPH'97, 1997

4. Sloan P P, Kautz J, Snyder J. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. In: Proc SIGGRAPH'02, 2002, 527-536

（鲍虎军　华炜）

shiti lianxi moxing

实体联系模型（entity-relationship model）

通过实体型及其间的联系型来反映现实世界的一种数据模型。它是一种用于描述应用需求的数据模型，又称 E-R 模型。实体联系模型可以形象地用图形表示，称为**实体联系图**。实体联系模型是由 Peter, P. S. Chen 于 1976 年提出的，广泛适用于软件系统设计过程中的概念设计阶段。

实体联系模型的基本语义单位是实体和联系。实体代表现实世界中客观存在的事物。一个实体由若干属性表示，实体的每个属性对于该实体有一个取值，这些取值可将这个实体区别于其他实体。具有相同属性的实体构成一个实体型。联系表示两个或多个实体间的关联。相同类型的联系构成一个联系型。

在实体联系图中，矩形框代表实体型；菱形框代表联系型；椭圆形框代表实体型和联系型的属性，相应的名字记入框中。联系型及其涉及的实体型之间以线段连接，并在线段的端部标注联系的种类（1∶m，m∶n 或 1∶1）。

下面是表示某工厂物资管理概念模型的 E-R 图的例子。物资管理涉及的实体型有：仓库、零件、供应商、项目、职工等。这些实体型之间的联系如下：①仓库和零件间具有多对多（m∶n）的联系，即一个仓库中可以存放多种零件；一种零件可以存放在多个仓库中。②仓库和职工间具有一对多（1∶m）的联系，即一个仓库可以有多个职工担任保管员，而每个职工只能在一个仓库工作。③职工之间具有一对多（1∶n）的联系，即仓库主任领导若干保管员。④供应商、项目和零件三者之间有多对多（m∶n∶p）的联系。

描述以上实体型及联系型的 E-R 图如图 1 所示。

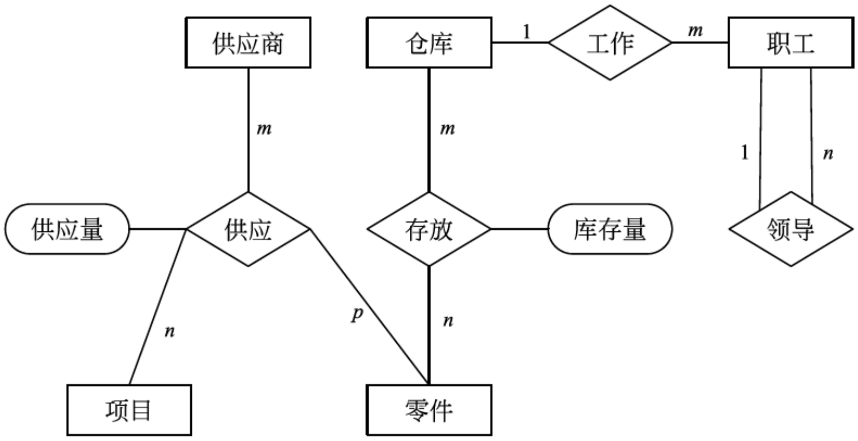


图 1　E-R 图示例

参考文献

Yao S B. Principles of database design. Vol. 1, Logical Organization. Englewood Cliffs, NJ: Prentice-Hall,1985 （王鹏 施伯乐）

shiti lianxi tu

实体联系图（entity relationship diagram）

用图形来描述概念模型的各实体间的联系的一种表示方法。又称ER图。现实世界的概念模型有很多表示方法，其中最常用的是P. P. S. Chen于1976年提出的实体联系模型（参见**实体联系模型**）。ER图是实体联系模型的图形表示，常用于数据库设计中的概念建模，亦可用于其他软件设计的建模。ER图提供了表示实体、属性、实体之间联系的方法。

实体：用矩形表示，矩形框内写明实体名。

属性：用椭圆形表示，并用无向边将其与相应的实体连接起来。

联系：用菱形表示，菱形框内写明联系名，并用无向边分别与有关实体连接起来，同时在无向边旁标上联系的类型。实体之间的联系有一对一、一对多、多对多的联系，分别用1∶1、1∶n、m∶n来表示。

参考文献

萨师煊，王珊. 数据库系统概论. 3版. 北京：高等教育出版社，2000 （李宣东 王珊）

shiwu chuli

事务处理（transaction processing） **数据库管理系统**中管理和协调事务以确保其正确执行的机制和过程。事务是完成单一逻辑功能的一个操作序列。事务处理部件必须保证事务执行满足ACID特性。其中，A（atomicity）表示原子性，含义是事务的所有操作或者全部执行或者全部不执行；C（consistency）表示一致性，含义是事务将数据库从一个一致状态转变为另一个一致状态；I（isolation）表示隔离性，是指尽管数据库系统中可以存在多个并发事务，但每一个事务都感觉不到系统中有其他事务在运行。在它看来，任何一个其他事务或者在它之前执行，或者在它之后执行；D（durability）表示持久性，指的是一个事务成功提交后，即使出现各种软硬件故障，它对数据库的改变必须是永久的。

事务的原子性和持久性由数据库管理系统的恢复管理部件来实现。恢复管理部件基于日志、影子数据库等技术，将数据的操作记录保存到稳定存储器中，确保在软硬件故障的情况下将数据库恢复到失败事务启动之前的状态以及实现永久保存成功事务对数据库的修改。

事务的隔离性由数据库管理系统的并发控制部件提供支持。多事务并发控制需要保证事务并发调度的执行结果等价于某个串行调度的执行结果，同时提高事务系统的吞吐量，减少单个事务的延迟。并发控制大多通过每个事务必须遵守的协议来实现。最常用的并发控制协议是基于封锁的协议，一个事务通过持有数据项上的锁延迟其他事务冲突指令的执行。此外，还有一些乐观的并发控制协议，如基于有效性检查的协议、基于时间戳的协议等。这些协议往往通过特定事务的回滚产生可串行化的并发调度。

事务的一致性确保随着事务的运行，数据库从一个一致状态转换到另外一个一致状态。数据库状态的一致性要求和应用相关。数据库管理系统通过**数据完整性**约束机制为单个事务数据一致性提供了一定程度的支持。同时，数据库系统保证多个事务并发执行不会破坏数据库状态的一致性。

当前商业**关系数据库**管理系统一般都能很好地支持事务处理。

参考文献

1. Silberschatz A, Korth Henry F, Sudarshan S. 数据库系统概念. 5版. 杨冬青，马秀莉，唐世渭，等译. 北京：机械工业出版社，2006

2. Gray J, Reuter A. 事务处理：概念与技术. 孟小峰，于戈，等译. 北京：机械工业出版社，2004

（杨冬青）

shiwu chuli zhongjianjian

事务处理中间件（transaction processing middleware） 一种用于产生、执行、协调和管理事务，为面向事务的应用提供良好编程环境和运行平台的**分布计算中间件**。

事务概念最早来源于**数据库管理系统**，是指具有原子性、一致性、隔离性和持久性（AC I D）四个事务特性的原子操作序列，被用来确保应用程序对数据库访问的一致性和可靠性。在早期应用程序中，商用数据库管理系统内部集成的事务管理器提供应用程序所需的事务处理功能。随着网络技术的发展以及应用需求的变化，以往集中式应用演化发展为网络分布式应用，数据和处理分布在不同的计算机上，事务处理技术需要解决分布性带来的问题。

此时，事务管理功能通常由专门的中间件提供，事务处理技术也发展为分布式事务处理。

事务处理中间件是保证应用程序正确、可靠运行的基础设施。在分布式环境下，事务（全局事务）需要在分布的不同计算机上执行，而每一台计算机只能保证其上事务（局部事务）的事务特性。事务处理中间件按照一定的事务处理协议负责协调各个局部事务的执行，并管理全局事务，以保持应用程序的事务特性。典型的分布式事务处理协议是两阶段提交协议。**并发控制**和失败恢复是事务处理的两个核心问题。并发控制用于保证多个并发执行事务的最终执行效果等价于某种串行执行的效果，即保证执行的串行等价性。目前常用的并发控制方法包括锁机制、乐观并发控制和时间戳机制等。失败恢复用来处理事务执行过程中可能出现的各种故障，目前常用的失败恢复方法包括日志和阴影页等。

X/Open DTP 参考模型为分布式事务处理系统提供了一个标准的软件体系框架，不同的应用程序之间能够共享由各种资源管理器提供的资源，并保证系统内各种资源均衡有效地在全局事务中协同工作。X/Open DTP 模型主要由应用程序、资源管理器和事务管理器等组成。其中，应用程序是指与具体业务相关的程序，定义了事务边界和组成事务的各种行为；资源管理器负责管理资源，为应用程序提供数据的插入、修改和删除等操作，并实现对资源的访问和并发控制，保证数据的完整性与一致性；事务管理器负责管理事务和资源管理器，对事务进行监控，完成事务提交和事务失败时的错误恢复，并负责资源管理器之间的调度、分配与协调。DTP 模型还定义了这三者之间的交互接口，其中应用最广泛的是资源管理器和事务管理器之间的 XA 接口以及应用程序和事务管理器之间的 TX 接口。

参考文献

1. 冯玉琳，黄涛，金蓓弘. 网络分布式计算与软件工程. 2 版. 北京：科学出版社，2011

2. Gray J, Reuter A. Transaction processing: concepts and techniques. Morgan Kaufmann, 1993

3. X/Open Company Ltd. Distributed Transaction Processing: Reference Model, Version 3, 1996

（魏峻）

shiwuyuan

事务元（transaction） 完成单一逻辑功能的、对数据库的一组有序操作（可以是增加、删除、修改、查询和存储等操作的各种组合）。事务元简称为**事务**。事务由事务开始（begin transaction）和事务结束（end transaction）之间执行的全体操作组成。事务的执行需要满足原子性、一致性、隔离性和持久性的要求。在整个事务生命周期中，事务在启动和运行过程中的状态称为活动状态；事务在最后一条语句执行后，日志工作结束前处于部分提交状态；如发现正常的执行不能继续，事务处于失败状态；事务回滚并且数据库恢复到事务开始之前的状态后，事务处于中止状态；事务成功完成后，处于提交状态。

参考文献

Silberschatz A, Korth Henry F, Sudarshan S. 数据库系统概念. 6 版. 杨冬青，李红燕，唐世渭，等译. 北京：机械工业出版社，2012

（何新贵）

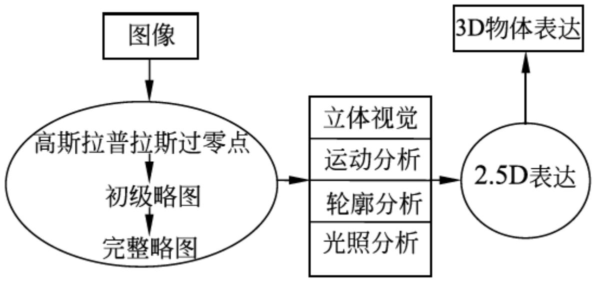
shijue jisuan lilun

视觉计算理论（computational theory of vision） 通过建立人类视觉功能的数学模型及其解的算法实现来阐明视觉具有可计算性的原理。视觉计算理论的奠基者 D. Marr 在其"Vision"一书中提出了视觉计算理论和方法。马尔计算视觉理论的提出，标志着计算机视觉成为了一门独立的学科。

D. Marr 计算视觉理论包含两个主要观点：首先，D. Marr 认为人类视觉的主要功能是复原三维场景的可见几何表面，即三维重建问题；其次，Marr 认为这种从二维图像到三维几何结构的复原过程是可以通过计算完成的，并提出了一套完整的计算理论和方法。所以，Marr 视觉计算理论文献中也被称为三维重建理论。

Marr 认为，从二维图像复原物体的三维结构，涉及三个不同的层次。首先是计算理论层次，也就是说，需要使用何种类型的约束来完成这一过程。Marr 认为合理的约束是场景固有的性质在成像过程中对图像形成的约束。其次是表达和算法层次，也就是说如何来具体计算。最后是实现层次。Marr 对表达和算法层次进行了详细讨论。他认为从二维图像恢复三维物体，经历了三个主要步骤，即图像初始略图（sketch）→物体 2.5 维描述→物体 3 维描述。其中，初始略图是指高斯拉普拉斯滤波图像中的过零点（zero-crossing），短线段，端点等基元特征。物体 2.5 维描述是指在观测者坐标系下对物体形状的一些粗略描述，如物体的法向量等。物体 3 维描述是指在物体自身坐标系下对物体的描述。如球体以球心为坐标原点的表述。Marr 三维重建理论可以

用下面的简图概述：



Marr 视觉计算理论是 20 世纪 80 年代初提出的。之后 30 多年的研究中，人们发现 Marr 理论的基本假设："人类视觉的主要功能是复原三维场景的可见几何表面"基本上是不正确的，"物体识别中的三维表达的假设"也基本与人类对物体识别的神经生理机理不相符。尽管如此，Marr 计算视觉理论在计算机视觉领域的影响是深远的，他所提出的层次化三维重建框架，至今是计算机视觉中的主流方法。尽管文献中很多人对 Marr 理论提出了质疑、批评和改进，但就目前的研究状况看，还没有任何一种理论可以取代 Marr 理论，可以与其相提并论。

参考文献

Marr D. Vision：A computational investigation into the human representation and processing of visual information，W. H. Freeman and Company，1982

（胡占义）

shipin bianma

视频编码（video coding） 将一种视频格式的数据流通过编码转换成另一种视频格式的数据流。通常情况下，视频编码的目的是对视频信号进行数据压缩。

传统的视频压缩编码是建立在香农（Shannon）信息论基础上的，用统计概率模型来描述信源，开发出预测编码、变换编码、矢量量化编码、子带-小波编码和神经网络编码等方法。

新一代视频压缩编码考虑了信息接受者的视觉特性、事件本身的含义和重要程度等因素，称为基于内容的压缩编码；其中基于对象（object-based）图像压缩方法和基于语义（syntax-based）的图像压缩方法成为研究热点。

常用的视频压缩编码标准有国际电联（ITU-T）的 H. 26X 系列标准，国际标准化组织（ISO）下属的运动图像专家组的 MPEG 系列标准，此外，在互联网上被广泛应用的还有微软公司的 WMV 以及 Apple 公司的 QuickTime 等标准。

以上视频压缩编码标准和技术大量用于视频光盘（VCD，DVD 等），可视电话，数字电视传输、IPTV 互联网电视，网络流媒体，视频安防监控，网络视频聊天和网络视频会议等领域。

参考文献

胡道元. 计算机网络（高级）. 北京：清华大学出版社，1999

（周杰　张凌）

shipin huiyi

视频会议（video conferencing） 集话音、文字和图像通信于一体的，用于多点实时交互式通信的一种多媒体通信技术。它可以将不同地点与会人员的活动情况、会议内容及各种文件资料以可视的形式展现在各个分会场。这是一种快速高效、应用日益增长的新的通信业务。

根据通信节点的数量，视频会议系统可分为：点对点视频会议系统和多点视频会议系统。

点对点视频会议系统支持两个通信节点间的视频会议通信功能，例如可视电话系统、分散在两个场地中的桌面视频会议系统及会议室型视频会议系统等。后者在会议室型视频会议系统的支持下，一群与会者集中在一间特殊装备的会议室中，与远地另一个会议室的与会人员进行交互通信，实现两点间的视频会议功能。由于会议室与会者较多，因此对视听效果要求较高，一套典型的系统一般应包括一台或几台大屏幕监视器、高质量摄像机、音响设备和控制设备等。

多点视频会议系统允许 3 个或 3 个以上不同场地的参加者同时参与会议。多点视频会议系统的一个关键技术是多点控制问题，多点控制单元（MCU）在通信网络上控制各个点的视频、音频、通用数据和控制信号的流向，使与会者可以接收到相应的视频、音频等信息，维持会议正常进行。

视频会议系统主要由视频会议终端、多点控制单元、信道（网络）及控制管理软件组成，见图 1。

视频会议终端组成主要有：音视频输入/输出单元、音视频压缩编码和解码单元，符合相应网际标准的信息处理和通信协议软件。

多点控制单元（multipoint control unit，MCU）的主要功能是对视频、语音及数据信号进行切换，MCU 的主要组成部分是：网络接口单元、呼叫控制单元、多路复用和解复用单元、音频处理器、视频处理器、数据处理器、控制处理器、密匙处理分发器及

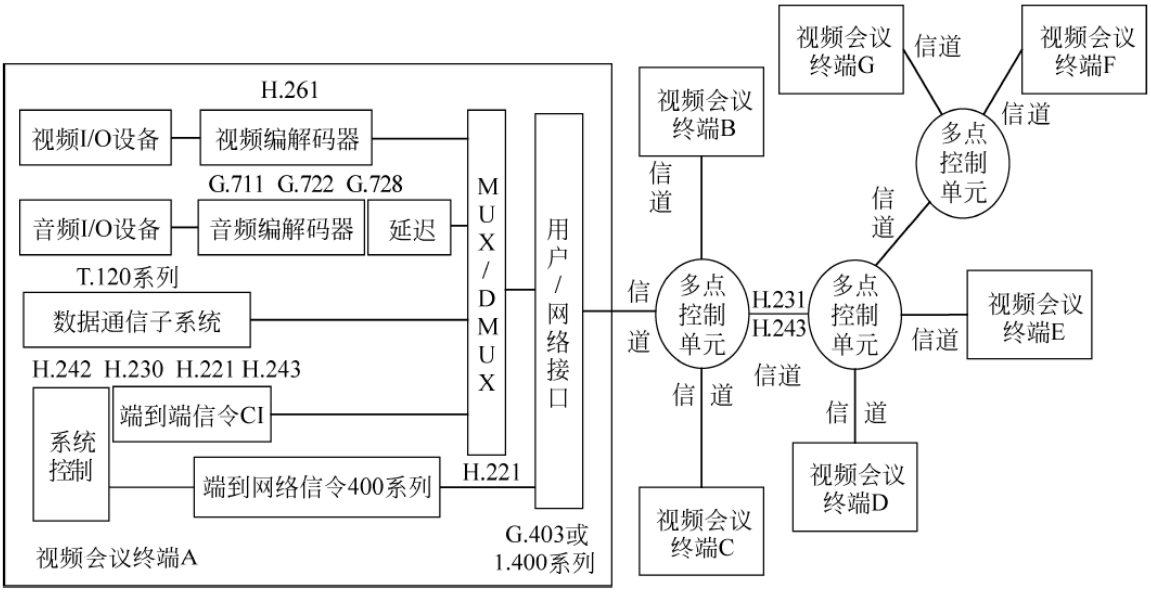


图 1　视频会议系统系统结构框图

呼叫控制处理器。

视频会议系统的服务质量(quality of service, QoS)是通过把用户的服务请求映射成 QoS 参数,再通过资源的分配和调度,满足 QoS 参数,进而满足用户的服务需求。资源的分配和调度管理有:①资源的静态管理,包括 QoS 的协商和解释,资源许可(admission),资源的保留和分配及资源释放;②资源的动态管理,包括进程管理、缓冲区管理、传输率和流量控制及差错控制。

视频会议的安全保密系统主要由加密模块和解密模块组成。加密模块是将会议终端用户数据加密送到网上传输,解密模块对加密数据进行解密得到用户数据。加密和解密的核心是密匙的生成和管理。密匙生成的核心是加密算法,加密算法不包含在网际标准的建议中,是由视频会议系统设计者自行研制和选用。

参考文献

钟玉琢,蔡莲红,史元春,沈洪. 多媒体计算机技术基础及应用. 3 版. 北京:高等教育出版社,2009

(钟玉琢　孙立峰)

shipin tuxing suiji cunqu cunchuqi xinpian

视频图形随机存取存储器芯片(video graphic random access memory chip)　用于存储视频、图形及文字信息的显示和处理数据的一种专用**随机存取存储器芯片**,它与视频或图形处理器直接相连,简称 VGRAM 芯片。VGRAM 芯片内不仅有存储信息的随机存取存储器芯片,而且可有适合于图形数据存储和显示的逻辑控制电路。

视频和图形显示是 PC、消费电子、移动通信等数字系统人机交互的必备手段。一幅图像由待显示的像素的阵列构成,其列数 × 行数定义为显示设备的分辨率。每个像素分为不同强度的红、绿、蓝三个基本色,强度的不同形成不同的颜色。由于显示设备的限制,像素阵列需定时刷新。一个完整的像素阵列称为一帧,这一帧像素的数据就存储在 VGRAM 中,用于图像显示。随着人们对视觉感受要求的提高,对显示设备的分辨率、每秒的帧数和每个像素表示基本色强度的数据位数也不断提高。除实时读出再生刷新图像数据外,还需不断输入修改图像数据,以形成动态变化的图像。这样仅为维持人能感受稳定清晰连续变化的图像,就要求存储视频图像的存储器有很高的数据传输率。

早期由于非同步动态随机存储器(DRAM)的数据传输率较低,出现了专用的视频随机存储器(VRAM)芯片,包括帧存储器芯片和多端口视频随机存取存储器芯片两种。

帧存储器芯片主要用于数字电视和录像机,由存储一帧图像的像素颜色表地址码的 DRAM 和存储一行像素颜色表地址的行缓冲寄存器(SRAM)组成。

多端口视频随机存取存储器芯片多用于计算机视频图像显示系统中,一般使用的是双端口 DRAM。两个端口分别为可随机访问的 DRAM 端口和可串行访问的串行存取存储器(SAM)端口。

随着3D电子游戏、数字影像、计算机辅助设计等应用对图像处理和显示要求的发展，出现了专用于2D/3D图形显示处理的图形处理器（GPU）芯片（参见**图形处理器**）。由于GPU芯片采用多路并行结构加速图形处理和绘制过程，其图形处理效率远高于通用CPU芯片，它对访存速率的要求也超过了普通的DDR存储器芯片（参见“**动态随机存取存储器芯片**”）。为了保证GPU芯片的高效率，出现了专用于GPU的图形双倍数据速率（DDR） DRAM（简称为GDDR）芯片。GDDR芯片的数据输出为32位（记为GDDR）或16位（记为gDDR），有多种不同容量和数据传输率的产品。表1所示为实际使用的GDDR芯片的特性。

最初的GDDR产品为8 M×16的芯片，其时钟频率高于同期的双倍数据速率（DDR）芯片，最高到150 MHz。与DDR芯片相同，时钟输入改为差分输入，以减小外部干扰产生的时钟抖动。

GDDR 2开始采用芯片内加匹配电阻（ODT）的技术，减小芯片外匹配电阻的分叉线带来的反射，有助于提升工作频率。

表1　GDDR芯片产品特性例

	GDDR 1	gDDR 2	GDDR 3 gDDR 3	GDDR 5
芯片容量构成	8 M×16 4 M×32	64 M×16	32 M×32 128 M×16	32 M×32（1 Gb） 64 M×32（2 Gb）
主时钟频率	100～150 MHz 100～200 MHz	400/500 MHz	0.7～1.3 GHz 0.7～1.3 GHz	0.8～1.5 GHz 写入时钟1.6～3.0 GHz
最高数据传输率	200～300 Mb/s 200～400 Mb/s	800 M，1.0 Gb/s	1.4～2.6 Gb/s 1.4～2.6 Gb/s	3.2～6.0 Gb/s
电源电压	2.5 V	1.8 V	1.8 V， 1.5 V	1.6 V（6 Gb/s），1.5 V， 1.35 V（低功耗3.2 Gb/s）
芯片内存储体数（Bank）	4	8	8	8/16（分四组）
封装和引脚数	TSOP 66（×16） FBGA 144	FBGA 84	FBGA 136 FBGA 96	FBGA 170

GDDR 3除了不断改进工艺技术外，在传输匹配技术上，也有新的改进，如控制芯片输出驱动器输出阻抗的OCD技术；在芯片输入端只有上拉匹配电阻的OCD和对应输出为伪漏极开路（POD）技术。

GDDR 4芯片采用了8位预取和数据反相（数据输出高电平为1）等新技术，但由于容量、速度和成本没有GDDR 3有吸引力，反而未得到大量应用。

GDDR 5芯片最突出的新技术是其频率为主时钟频率的二倍的差分输入写入时钟。这二对差分时钟分别控制双向数据端口。通常无专门的读出时钟输入，内部确定读出数据的时钟由输入的写入时钟经锁相环调整延迟时间产生。数据传输率为写入时钟频率的二倍，为输入时钟频率的四倍，即主时钟频率0.8～1.5 GHz，数据传输率可达3.2～6.0 Gb/s，差分时钟的频率为1.6～3.0 GHz。为了能在这样高的数据传输率下可靠工作，除了继续使用ODT、OCD和POD技术外，在减少电源和地的噪声对时钟和数据信号的影响和减小功耗上也采取了很多措施。

GDDR5芯片已在**个人计算机**（PC）和工作站系统的高/中端显卡中广泛采用一些GPU芯片的显存接口字长有128～384位，总容量512 MB～6 GB，最多可支持6台2600×1600的显示器，以满足三维立体影像、计算机辅助设计（CAD）的复杂图像、影像创造和编辑等领域的需求。

参考文献

1. Sharma A K. 先进半导体存储器——结构、设计与应用. 曾莹，等译. 北京：电子工业出版社，2005

2. http://www.samsung.com/Products/Semiconductor

（孙祖希）

shoushi shibie

手势识别（hand gestures recognition）　计算机识别人手（或手和臂的组合）姿态或动作并判断

其意义的技术。人手(或者手和臂的组合)的各种姿态或动作称为手势,包括静态手势(手的单个姿态、手形)和动态手势(动作,由一系列姿态组成)。

基于视觉的手势识别其基本原理如图 1 所示。系统通过输入设备获取图像序列(或视频流),检测其中是否有手势出现。如果有,则把该手势从图像序列(或视频流)中分割出来,然后进行手势分析,选择手势模型并获取模型参数,最后对手势进行识别、分类形成关于手势的描述,从而得到最终的识别结果。

手势模型的建立(手势建模),对手势进行分析确定模型参数(手势分析)以及最终的手势识别等是手势识别的 3 个重要组成部分,也是手势识别的技术关键。

手势模型对于手势识别至关重要,是指通过计算机提取出来的可以用来反映人手(或手和臂的组合)的静态表现(姿态)及动态运动特征(动作)的抽象化的模型。手势模型可以分成基于三维手(臂)模型的手势模型和基于表观的手势模型两大类,其中前者包括纹理模型、网格模型、几何模型以及骨架模型,后者包括基于灰度图像本身的表观模型、基于二维变形模板的模型、基于图像属性的模型、基于图像运动的表观模型等。图 2 举例给出了表示同一姿态的各种人手模型,可以在此人手模型的基础上建立相应的手势模型。

手势建模　指通过分析手势在图像(序列)中的表观姿态或动态运动特征并建立与其相对应的手势模型的过程,其目的是对表示各种不同意义的手势分类并建立模型。

手势分析　任务是估计选定的手势模型的参数,包括特征检测和参数估计两个部分。特征检测通过基于颜色定位、基于运动定位以及多模式混合定位等技术手段检测图像(序列)中的人手位置;然后选择手势模型并估计模型的参数。

手势识别　根据手势分析得到的模型参数(序列)识别其中的含义(意义)并得到关于手势描述的识别结果。根据静态手势识别或动态手势识别,其技术方法也有所不同。静态手势识别算法主要包括基于经典参数聚类技术的识别、基于非线性聚类技术的识别等;动态手势识别由于涉及时间和空间上下文的问题,因而比较复杂,常用的识别技术有基于隐马尔可夫模型(HMM)、基于动态时间规整(DTW)以及基于压缩时间轴的识别等。

手势是一种自然直观的人际交流方式,利用计算机自动进行手势识别将大大增强现有的人机交互技术,实现更直接、更自然、更和谐的人机交互接口。手势识别在计算机远程操作、交互游戏、噪声环境中的交互交流等都会有广泛的应用。

手势识别的研究工作开始于 1992 年。近年来,手势识别技术获得了研究人员的普遍重视,国内外一些著名的研究机构和学校等都开展了这方面的工作,目前已经取得了一些研究成果。然而,手势识别也还存在一些问题,比如光照、背景的影响等。另外,成功的手势识别策略应该考虑手势的时间和

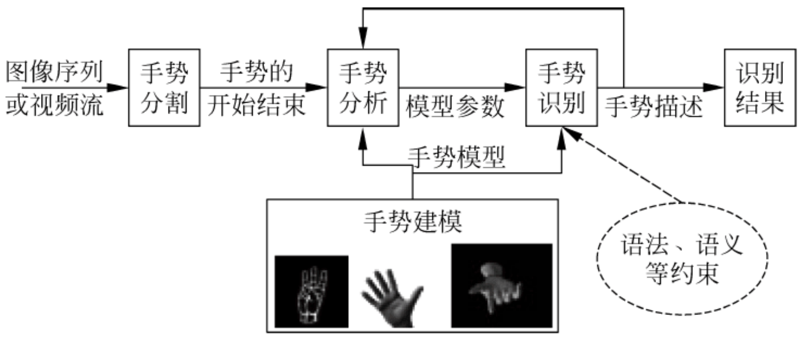


图 1　基于视觉的手势识别基本原理框图(图片引自参考文献 1)



(a) 有纹理的3D体模型　(b) 3D网格模型　(c) 3D骨架模型　(d) 二值影像模型　(e) 轮廓模型

图 2　表示同一姿态的各种人手模型(图片引自参考文献 1)

空间上下文,即考虑手势的语法规则。语法规则既要反映手势的语言学特征,又要反映手势的空间特征。手势识别还处于研究阶段,离真正进入实际应用还有较长的路要走。

参考文献

1. 任海兵, 祝远新,徐光祐,等. 基于视觉手势识别的研究综述. 电子学报,2002, 28(2):118-121

2. Hamdam R, Heitz F, Thoraval L. Gesture localization and recognition using probabilistic visual learning. In:Proceedings of the IEEE Comp Soc Conference on Computer Vision & Pattern Recognition, 1999, (2):98-103

(蔡莲红　吴志勇)

shouxie shuru ban

手写输入板(handwriting input board)　用于输入文字的一种**输入设备**。其作用类似**键盘**,也兼具一些绘画和**鼠标**的功能。

手写输入板一般使用一支专门的笔,或者手指在特定的区域内书写文字。手写输入板通过各种方法将笔或者手指的轨迹记录下来,然后通过文字识别软件转换成文字(参见**联机手写汉字识别**)。对于不熟悉中文输入法的人来说,只要会写字便可方便地将文字输入计算机。

手写输入板有的集成在键盘或鼠标上,有的是自成一体,自成一体的手写输入板一般使用 USB 口、串口或者无线连接与计算机相连。目前手写输入板种类很多,有兼具手写输入汉字和光标定位功能的,也有专门用于屏幕光标精确定位以便完成各种绘图功能的。

手写输入板按工作原理可分为以下几类:

(1) 电阻压力式　手写输入板由一层可变形的电阻薄膜和一层固定的电阻薄膜构成,中间由空气相隔离。当用笔或手指接触手写输入板时,利用接触电阻受压变形感应出笔或手指的位置。

(2) 电容触控式　手写输入板通过人体的电容来感知手指的位置,在触控板表面附着有一种传感矩阵,用以持续不断地跟踪使用者手指电容的"轨迹",经处理,精确定位每个瞬间手指的位置(X、Y 坐标),手指与板间距离(Z 坐标),最终完成 X、Y、Z 坐标值的确定。因为电容式触控板所用的手写笔无须电源供给,特别适合于便携式产品。

(3) 电磁压感式　手写输入板通过下方的布线电路通电后,在一定空间范围内形成电磁场来感应带有线圈的笔尖,从而确定笔尖的位置进行工作。新一代的手写输入板又引入了压力感应技术,笔尖可以随着用力的大小微微的伸缩,一个附加的传感器能感应到笔尖上所施加的压力,并将压力值传给计算机,计算机则在屏幕上显示出该值笔迹的粗细。这样手写输入板完全就可以同真实的笔相比拟了,除了手写文字输入以外,还可以用来画画或模仿毛笔泼墨。市场上常见的压感电磁板有 256 级压力和 512 级压力两种。

上述三类手写输入板的性能、耐用性和价格等方面各有优缺点,电阻板技术最为古老,而电容板由于手写笔无须电源供给,多应用于便携式产品。电磁板则是目前最为成熟的技术,可以进行流畅的书写,手感也很好,还可分辨笔迹粗细,已经被市场所认可,应用最为广泛。

随着处理器性能的不断提高以及汉字识别软件日趋成熟,手写汉字识别率、易用性和实时性均达到了实用程度,联机汉字输入已获得市场认可。联机手写输入这种人机通信方式的简捷、人性化等特点,为计算机的普及应用和手写输入展示了良好的前景。

(韩承德)

shuchu shebei

输出设备(output device)　将信息处理系统(如计算机)处理过的数据信息形式转换为人能够识别和理解的信息形式的设备。这些形式包括文字、图形、图像、视频和声音等。常见的输出设备有**显示器**、**投影仪**、印刷设备、**绘图机**和声音输出设备。显示器和声音输出设备的输出结果不能长期保存,而印刷设备和绘图机可用来长期保存文字、图形和图像的输出结果。

目前,显示器常见的有阴极射线管(CRT)显示器、液晶显示器和等离子显示器;而新兴的显示器有电子纸、发光二极管显示器、3D 显示器、**头盔显示器**等。投影仪是将计算机的数据转换为图像或视频后投影到幕布或墙面上以显示信息的设备。

印刷设备是一种提供硬拷贝的计算机输出设备,它有许多种类型。根据印字头对纸有无击打动作可分为**击打式打印机**与**非击打式印刷机**两大类;按字形的表现方法可分为完整笔画的整字型印刷与用紧密邻接的点构成字形的点矩阵(点字符)型印刷两类;按印刷过程中逐字处理与逐行处理的方式又可分成串行印刷与并行(行式)印刷,以页为单位进行印刷的称为页式印刷。按颜色分有单色印刷与彩色印刷。

早期的计算机印刷输出设备基本上是利用机电原理的击打式打印机。这类设备的机械结构复杂,打印速度受机械的限制难以提高,且噪声大,存在机械磨损问题。击打式的优点是可使用普通纸,并可同时复印数份。非击打式印刷设备利用物理的(光、电、磁、热)或化学的方法实现印刷,而不是靠机械冲击的方法,因而噪声低,速度快。常见的有激光印刷机、喷墨印刷机、静电印刷机、热升华印刷机等。其缺点是不能同时印刷多份,有些非击打式印刷设备还需用特殊的印刷纸。非击打式印刷机几乎都是点字符型,字型、字体变换灵活,适于图形、图像及汉字的印刷。

声音输出设备将代表声音(语言和音乐)的数字信号通过数模转换变换为模拟信号,放大后送到扬声器(由电磁铁、线圈、喇叭膜组成)或耳机将其转化成人耳可以接听的声波。

另外,将计算机的电子信息转变成其他机器所能识别(感知)的形式的设备也可以认为是一类输出设备。例如联机的卡片穿孔机、纸带穿孔机、**磁带机**、软盘驱动器、光碟驱动器等均可视为这类输出设备。例如,通过联机**磁带机**或**软磁盘驱动器**等存储设备暂时将待输出的信息记录在磁带或软磁盘等媒体上,然后将这些媒体转移到由磁带机或软磁盘驱动器控制的脱机打印设备或绘图机上,进行脱机印刷或绘图。

计算机输出的数字信号经**数模转换器**转换为模拟信号,再送往自动控制系统进行过程控制。数模转换器等亦可视为一类输出设备。

(刘锡刚　黄建忠)

shuru shebei

输入设备(input device)　将待输入的数字信息或各种其他形式的信息转换成适宜于计算机处理的数字信息并送入计算机的设备。输入设备可分为两类:采用媒体输入的设备和人机交互输入设备。

采用媒体输入的设备有纸带输入机、卡片输入机、光学字符阅读机(OCR)、光学标记阅读机(OMR)、**扫描仪**、**磁带机**、**软磁盘驱动器**、**磁光碟驱动器**等。这些设备把记录在各种媒体(如:穿孔纸带、穿孔卡片、磁带、软磁盘、光碟、移动硬盘、U 盘、纸张、信息卡)上的信息送入计算机。一般是成批输入,输入过程中使用者不作干预。

人机交互输入设备不使用媒体,而由使用者通过操作直接和计算机进行信息交换。常见的有**键盘**、跟踪球、**鼠标**、**触屏**、数字化仪、**手写输入板**、光笔等。人机交互输入设备通常和显示器连用。使用者在输入时,从显示器的屏幕上可以看到输入的内容和计算机对输入所作出的反应,即输入是采用使用者和计算机对话的方式进行的。如用键盘编辑、修改程序,用**控制杆**玩电子游戏,用鼠标操作、控制计算机等都是交互输入。在输入过程中,使用者可以即时进行处理,如保留、删去、插入、修改、移动等。交互输入一般不强调快速性,而重视人机界面的友善性,即使用方便,对话自然等。

20 世纪 90 年代以来,许多自然语言输入设备,如言语(语音)输入设备、文字输入设备和图像输入设备等,都有商品问世。

言语(语音)输入设备是将人类的言语转换为计算机能接受的数字信号并加以识别的设备。言语输入设备由以下几个部件组成:①话筒等拾音器将言语转换为模拟电信号。②模数转换器将经过放大的模拟电信号转换成数字信号。③计算机和言语识别软件,将以数字表示的言语信息与计算机内部所存储的言语模型进行比较,从而找出最佳匹配作为识别结果。这部分是言语输入设备的核心。言语输入设备有广阔的应用领域,如言语听写机(言语输入代替键盘输入)、声控系统(使用声音进行自动控制)、电话的语音拨号(以说人名或单位名代替拨号)等(参见**汉语言语识别**)。

文字输入设备在我国主要是汉字识别系统。20 世纪 70 年代末,我国开始对**汉字识别**进行研究。20 世纪 90 年代以来技术逐步成熟,已形成商品投入市场。

汉字识别有两种类型:①**联机手写汉字识别**。用特种笔或手指在手写板上写字,一边写一边认,是一种交互型汉字输入手段。设备由手写板和**微型计算机**组成。②**印刷体汉字识别**。识别印刷在纸上的各种印刷体汉字,包括照排机、打印机输出的汉字和印刷的汉字。通常能识别宋、仿宋、楷、黑等各种印刷体,同时能识别西文和数字以及一些符号。印刷体汉字识别设备由扫描仪和微型计算机组成。

随着**多媒体技术**的发展,除**图像获取**之外,**数字音频获取**与**数字视频获取**都成了计算机的重要技术。因此,相应的设备,如扫描仪、**数字相机**、**数字摄像头**和数字摄像机等也都成了计算机的常用输入设备。

人机交互输入设备是研究的重要问题之一。因为人机交互能力是影响计算机使用的一个重要因

素。为了实现自然语言输入,使人与计算机之间能像人与人之间一样采用自然语言交谈,需要有更完善、更理想的交互输入设备。 (林兼)

shuru shuchu guanli chengxu

输入输出管理程序(input/output manager) 操作系统中用于组织和管理输入输出(I/O)设备,以完成I/O操作的程序。它的主要任务是有效地处理对I/O设备的使用请求、实现I/O程序设计和I/O设备驱动调度,实现主存和外围设备的数据传输操作。通常,I/O设备、I/O控制部件和I/O软件统称为I/O系统。

设备独立性是对设备管理的基本要求。实现设备独立性的关键是把用户所用的设备与计算机中实际进行I/O操作的物理设备分离开来,系统通过逻辑设备来接受用户的I/O请求,而由设备管理程序来实现由逻辑设备到物理设备的转换。

在I/O系统中,普遍采用I/O中断、缓冲区管理、通道等多种技术,较好地克服了由于I/O设备和处理器在速度上的不匹配所引起的缺点,使主机和外设并行工作,显著改善了I/O设备的使用效率和系统性能。

I/O软件分为四层:设备中断处理程序、设备驱动程序、独立于设备的I/O软件和用户层I/O软件。设备中断处理程序在底层,主要工作有:分析中断类型做出相应处理,检查和修改程序状态等;设备驱动程序中包括所有与设备相关的代码,主要工作是:把用户提交的逻辑I/O请求转化为物理I/O操作的启动和执行,如设备名转化为端口地址、逻辑记录转化为物理记录、逻辑操作转化为物理操作等;独立于设备的I/O软件的功能是执行适用于所有设备的常用I/O功能,并向用户层软件提供一致性接口,如设备命名、设备保护、缓冲管理、存储块分配等;用户层I/O软件包括在用户空间运行的I/O库函数和SPOOLing程序。

对于磁盘等作为后援存储器的大容量辅存设备,同时会有许多I/O请求到达,具有繁重的工作负载,为了降低若干个I/O请求执行的总时间,从而提高系统效率,应该按最佳次序执行这些I/O请求,这称作设备驱动调度,使用的算法叫驱动调度算法。有效的驱动调度算法除循环排序、优化分布、多重副本等外,还有:先来先服务、最短查找时间优先、电梯调度算法、扫描算法等。

为了实现设备的分配和去配,系统建立和维护一张设备状态表,表中每个表项应包括:设备类型、设备地址、占用进程、当前设备状态,请求使用设备的程序队列指针等。按设备使用方式可分为:独占设备、共享设备和虚拟设备。让一个程序整个运行过程中占用的I/O设备称独占设备,独占设备采用静态分配。共享设备允许多个用户同时使用,像大容量辅助存储器,一般不必进行分配,用户可通过文件系统按名存取共享设备中的信息。打印机可作为共享设备,这时采用动态分配,程序通过申请和(或)释放系统调用来获得I/O设备,系统则动态分给各程序使用。

虚拟设备所用技术称为假脱机操作技术,是一类物理设备模拟另一类物理设备的技术,是使独占设备变成共享设备的技术。其基本实现思想是:利用多道程序设计技术,在主机执行计算任务的间隙,将成批作业及其数据输入到磁盘称之为输入井的缓冲区中;此后,由高级调度程序选择作业(见**作业管理程序**)多道执行。作业使用数据时,不必再启动独占设备,而只要从相应输入井读取数据。同样地,作业执行中不必直接启动输出设备输出数据,而将作业的输出数据暂存到磁盘称之为输出井的缓冲区中,待程序执行完毕,由系统组织数据成批输出。实现SPOOLing系统的主要数据结构有:作业表、预输入表和缓输出表;SPOOLing系统的主要组成部分有:预输入程序、缓输出程序和输入井、输出井管理程序。

参考文献

孙钟秀,等. 操作系统教程. 4版. 北京:高等教育出版社,2008 (费翔林)

shuru shuchu jishu

输入输出技术(input/output technique) 实现计算机主机与外围设备进行信息交换的技术。**外围设备**包括**辅助存储器**和计算机与外部世界交换信息的输入输出设备。

输入输出设备根据其在计算机系统中的作用可分为:①**输入设备**主要有**键盘**输入设备、**鼠标**、**扫描仪**、**触屏**、纸带输入机、卡片输入机、光学字符识别设备(参见**光学字符阅读机**)、图形数字化仪(参见**图形输入设备**)、磁墨水字符识别设备和**语音输入**设备等;②**输出设备**主要有打印机、**绘图机**、磁盘机、纸带机、纸卡片穿孔机以及字符图形**显示器**等;③**终端设备**主要有会话型终端、智能型终端以及远程成批处理终端等;④**脱机设备**指没有计算机控制也可

完成数据编制、媒体变换等工作的设备,主要有卡片穿孔机、纸带穿孔机、磁带-绘图机等;⑤其他设备泛指在主机处理数据前后对数据进行加工的设备,包括**模拟输入输出通道**、外围处理机、开关量输入输出装置以及各种检测设备。另外,还有**通信控制器**,通过它可以实现计算机和通信线路的连接。

根据**中央处理器**输入输出操作参与的程度,计算机系统中**输入输出控制方式**有4类,即:①程序控制,包括无条件(同步)传送、条件(查询或异步)传送和中断传送;②**直接存储器存取**(DMA)控制;③通道控制;④输入输出处理机控制。

系统总线是计算机系统中各功能模块间传送信息的公共通路。各种**总线标准**的特性及应用也与输入输出技术密切相关。

参考文献

1. 王爱英. 计算机组成与结构. 4版. 北京:清华大学出版社,2007

2. 孙德文. 微型计算机技术. 3版. 北京:高等教育出版社,2010

（孙德文）

shuru shuchu jiekou

输入输出接口(input/output interface)　参见**输入输出设备接口**。

shuru shuchu kongzhi fangshi

输入输出控制方式(input/output control method)　指计算机主机对外围设备的管理方式。在计算机系统中输入输出是指计算机主机与外围设备进行数据传送。由于外围设备的工作速度差别很大,因此主机在同外围设备进行数据传送时,何时能用IN指令通过输入端口从输入设备读入数据以及何时能用OUT指令通过输出端口向输出设备输出数据,是一个复杂的定时问题。在主机同外围设备的数据传送过程中,要实现二者之间数据的正确传送,关键问题是主机对外围设备的管理方式,也就是数据传送的控制方式。在计算机系统中数据传送的控制方式有如下几种:①无条件传送方式;②程序查询方式;③程序中断方式;④直接存储器存取方式(DMA);⑤通道方式;⑥外围处理机方式。其中①、②、③属于程序控制方式,这三种控制方式和DMA方式是微型机系统和小型机系统中常用的控制方式,⑤、⑥两种一般用于大型机和服务器。程序控制的数据传送分为无条件传送、查询传送和中断传送,这类传送方式的特点是以中央处理器(CPU)为中心,数据传送的控制来自CPU,通过预先编制好的输入或输出程序(传送指令和I/O指令)实现数据的传送。这种传送方式的数据传送速度较低,传送路径要经过CPU内部的寄存器,同时数据的输入输出的响应也较慢。

1. 无条件传送方式

无条件传送方式又称"同步传送方式"。主要用于外设的定时是固定的且是已知的场合,外设必须在微处理器限定的指令时间内准备就绪,并完成数据的接收或发送。通常采用的办法是:把I/O指令插入到程序中,当程序执行到该I/O指令时,外设必定已为传送数据做好准备,于是在此指令时间内完成数据传送任务。无条件传送是最简便的传送方式,它所需的硬件和软件都较少。

2. 程序查询方式

程序查询方式又称"异步传送方式"。当CPU同外设工作不同步时,很难确保CPU在执行输入操作时,外设一定是"准备好"的;而在执行输出操作时,外设寄存器一定是"空"的。这样为保证数据传送的正确进行,提出了查询传送方式。当采用这种方式传送前,CPU必须先对外设进行状态检测。完成一次传送过程的步骤如下:① 通过执行一条输入指令,读取所选外设的当前状态。② 根据该设备的状态决定程序去向,如果外设正处于"忙"或"未准备就绪",则程序转回重复检测外设状态,如果外设处于"空"或"准备就绪",则发出一条输入/输出指令,进行一次数据传送。

3. 程序中断方式

为了提高系统的工作效率,充分发挥CPU高速运算的能力,在计算机系统中引入了"中断"系统,利用中断来实现CPU与外设之间的数据传送,这就是中断传送方式。

在中断传送方式中,通常是在程序中安排好在某一时刻启动某一台外设,然后CPU继续执行其主程序,当外设完成数据传送的准备后,向CPU发出"中断请求"信号,在CPU可以响应中断的条件下,现行主程序被"中断",转去执行"中断服务程序",在"中断服务程序"中完成一次CPU与外设之间的数据传送,传送完成后仍返回被中断的主程序,从断点处继续执行主程序。

采用中断传送方式时,CPU从启动外设直到外设就绪这段时间,一直在执行主程序,而不是像查询方式中处于等待状态,仅仅是在外设准备好数据传

送的情况下才中止 CPU 执行的主程序,在一定程度上实现了主机和外设的并行工作。同时,如果某一时刻有几台外设同时发出中断请求,CPU 可以根据预先安排好的优先顺序,按轻重缓急处理几台外设同 CPU 的数据传送,这样在一定程度上也可实现几个外设的并行工作。

4. 直接存储器存取方式

直接存储器存取方式(direct memory access, DMA)是一种不需要 CPU 干预也不需要软件介入的高速数据传送方式。由于 CPU 只启动而不干预这一传送过程,同时整个传送过程只由硬件完成而不需软件介入,所以其数据传送速率可以达到很高。在 DMA 传送方式中,对这一数据传送过程进行控制的硬件称为 DMA 控制器(DMAC)(参见**直接存储器存取**)。

5. 通道方式

通道是一个具有输入输出处理器控制的输入输出部件。通道控制器有自己的指令,即通道指令,能够根据程序控制多个外部设备并提供了 DMA 共享的功能。

6. 外围处理机方式

外围处理机(peripheral processing unit,PPU)基本上是独立于主机工作的,它有自己的指令系统,完成算术/逻辑运算、读/写主存储器,与外设交换信息等;有的外围处理机干脆就选用已用的通用机。外围处理机 I/O 方式一般应用于大型高效率的计算机系统中。

参考文献

1. 王爱英. 计算机组成与结构. 4 版. 北京:清华大学出版社,2007

2. 孙德文. 计算机组成基础. 北京:机械工业出版社,2009

3. 徐炜民,严允中. 计算机系统结构. 3 版. 北京:电子工业出版社,2010

(徐良贤)

shuru shuchu shebei jiekou

输入输出设备接口(input/output device interface) 计算机主机与外围设备之间以及两个外围设备之间的交接部分。在计算机系统中,计算机与外围设备之间应有特定的硬件连接和相应的软件控制,便于完成计算机与外围设备之间交换信息。这个硬件和软件的综合体称为输入输出设备接口,简称 I/O 接口,也称为外围设备接口。

随着大规模集成电路的发展,接口有专用芯片可供选用,使接口技术进入标准化阶段。另一方面,在许多情况下,采用**微处理器**作外围设备的接口芯片,接口向智能化方向发展。

输入输出接口的内容,涉及 I/O 接口的功能、I/O接口的连接方式和工作方式,以及 I/O 接口的类型等。

I/O 接口的功能

为了满足不同外围设备在不同速度、不同方式下的工作需要,接口应具有以下基本功能:

(1) 地址译码和设备选择　在系统中常有多个外围设备,所以系统有多个与之对应的接口。**中央处理器**(CPU)用地址码选择外围设备接口。因此接口本身就要有地址译码的能力。

(2) 数据缓冲和锁存数据　接口要将数据先锁存起来,以后再发送。解决主机与外围设备速度不匹配的矛盾。

(3) 信息变换　接口具有传送数据并行/串行转换的能力。当输入输出信号是模拟量时,要求接口具有模数(A/D)、数模(D/A)转换的能力。

(4) 传递控制指令和状态信息　主机和外围设备进行数据交换前,要确信设备是准备就绪的。因此,要对外围设备进行查询与控制,包括操作时序的协调、**中断**请求与响应、**直接存储器存取**(DMA)请求与响应、主机命令与**输入设备**和**输出设备**状态的交换与传送等。

I/O 接口的连接方式和工作方式　输入输出设备接口有多种连接方法,从主机的形式分,有**小型计算机**接口,**微型计算机**接口,**大型计算机**接口和分布式计算机接口。从接口结构及功能选择的灵活性分,有可编程接口和不可编程接口。从接口的数据传送方式分,有并行接口和串行接口。从接口的通用性分,有通用接口和专用接口。按输入输出的信号分,有数字接口和模拟接口。图 1 示出了主机、输入输出设备接口和外围设备之间的连接关系。

由图 1 可见,输入输出设备接口应有两个连接方向的接口。一个是主机与设备控制器之间的接口,它控制外围设备与主机总线之间交换数据,称之为系统级接口,如常见的小计算机系统接口(SCSI)、智能外围接口(IPI)、外围设备互连接口(PCI)等。另一个是设备控制器与设备驱动器之间的接口,它根据主机的命令管理外围设备的操作,称之为设备级接口。如存储模块设备接口(SMD)、智能设备接口(EIDE)、通用串行总线接口(USB)、IEEE-1394 接口和蓝牙接口等。

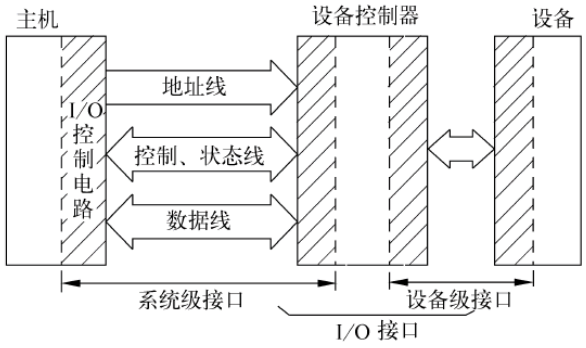


图 1　主机、输入输出设备接口及外围设备的连接方法

输入输出设备通过接口与主机进行信息传送的工作方式主要有四种:程序控制方式、程序中断方式、直接存储器存取(DMA)方式和输入输出处理机(通道)方式。

大、中型计算机因主机速度高,**操作系统**对主机资源的管理能力较强,其输入输出设备接口主要采用多路独立通道结构。

分布式计算机通过网络连接的外围设备的数量较多,它主要是利用多路通道的分时控制功能来分时处理和各外围设备交换数据。

在小型计算机和微型计算机中,输入输出接口一般多采用程序控制、程序中断和 DMA 三种控制方式。这些方式中接口的硬件都很简单,它们在不同的软件支持下,可以和多种外围设备连接,前两种方式主要用于低速设备,DMA 方式主要用于具有实时要求的外围设备和高速设备。

随着计算机技术的发展,输入输出设备接口技术已成为有效增强计算机效率与功能的重要环节。它的发展趋势是更加标准化、系列化和通用化。

I/O 接口的类型　按计算机与外围设备间的数据传输方式,可分为串行接口和并行接口两类。

串行接口　计算机与外围设备之间按顺序逐位进行数据传送的一种通信接口。

串行接口通信方式有两种:一种是异步通信方式。在这种方式中,一般以一帧为一个单位传送,一帧由起始位、数据位、奇偶检验位、结束位等部分组成。另一种是同步通信方式,发送端与接收端之间用同步信号进行数据通信。

常见的串行接口有 RS-232C、USB 接口、IEEE-1394 接口、蓝牙(bluetooth)技术、红外传输技术等。

(1) RS-232C 接口遵循常用的串行通信接口标准。它是 1969 年由美国电子工业协会(EIA)制定的。标准规定:不加调制解调装置时最长传输距离为 100 ft(1 ft = 0.3048 m)。信息传送速率最大为 19 200 Baud(1 Baud = 1 b/s)。RS-232C 采用 25 线的 D 型连接器。当不使用调制解调器直接连接时只要 5 根线。RS-232C 采用负逻辑。逻辑 0 为 +5 ~ +15 V,逻辑 1 为 −5 ~ −25 V。

(2) USB 接口　1994 年由 Compaq、HP、Intel、IBM 等公司提出的。主要特点有:自动检测即插即用,带电热插拔,支持菊花链式串行连接,一台计算机在一条通道电缆上可同时接入 127 个各种设备。

USB 采用专用连接器和电缆。有 4 根线,其中两个用于传输差分信号,两个用于给 USB 设备供电。USB 设备也可以由单独的电源供电。

USB 的传输类型有 4 种:①中断传输　由 USB 设备发出中断请求,传输的方向总是从设备到主机。②控制传输　用于配置 USB 设备并对它操作的某些方面进行控制。③同步传输　要求恒定传输速率的实时应用。如用于音频传输(如 CD-ROM)。④批量传输　用于对数据传输速率没有特殊要求的设备,如打印机。

USB 使用差分信号进行串行传输。2000 年制定的 USB2.0 标准规定最高数据传输速率是 480 Mb/s。2008 年制定的 USB 3.0 最高数据传输速率是 5.0 Gb/s。

USB 技术具有开放性。可用于台式机,也可用于便携机。可用于硬盘机(参见**硬磁盘驱动器**)、光碟机(参见**光存储器**)等存储设备,也可用于**鼠标**、**控制杆**、**键盘**、**显示器**、打印机及其他输入输出设备。在数字图像、语音合成、交互式多媒体、消费电子产品等领域也得到了广泛的应用。

(3) IEEE—1394 接口　Apple 公司和 TI 公司开发的高速串行接口。原型称 Fire wire。1995 年由 IEEE 定为 IEEE—1394 标准。IEEE—1394 接口应用在许多领域,如**数字相机**、数字摄像机、外接硬盘、打印机、**扫描仪**等设备。接口可以同时传送数字视频信号和数字音频信号,最高数据传输速率达 400 MB/s。它具有把一个信息源传来的数据向多个设备输出的功能,特别适合于家庭视听 AV 设备的连接。

每一个支持 IEEE-1394 标准的设备都有输入输出接口,用户可以采用方便的节点串联方式同时连接最多 63 台不同的设备。IEEE-1394 不要求个人计算机(PC)作为所有接入外设的控制器,不同的外设之间可以直接相互交换信息。另外,采用 IEEE-1394 接口,两台 PC 还可以共用同一个外设。这是

USB 或其他接口都做不到的。

IEEE-1394 插头座有 6 针和 4 针两种。6 针线缆中,两条双绞线用于数据传输,一条为地线,一条用于电源。4 针的没有电源和地线。线缆长度小于 4.5 m。

IEEE-1394 接口支持热插拔。

(4) 蓝牙(Bluetooth) Bluetooth 是公元 10 世纪一位国王的名字。1997 年爱立信公司用它命名一种短距离的无线电技术。利用“蓝牙”技术,手机与耳机之间不再需要连线;在个人计算机的主机与键盘、显示器和打印机之间也可以不用连线;现代的移动通信设备和计算机设备,不必借助电缆就能联网。在更大范围内,电冰箱、微波炉和其他家用电器可以与计算机网络连接,实现智能化操作。蓝牙技术的数据传输速率最高为 1 Mb/s,以时分方式进行全双工通信,通信距离为 10 m。

(5) 红外传输技术 红外传输技术是以红外线的方式传递数据。传输速度已达到 4 Mb/s(2011 年)。缺点是传输距离短,只能直线传输。

并行接口 计算机与外围设备之间进行并行数据传送的一种接口,具有在多条线上一次同时传输一组二进制位的能力,通常能将给定字节(或者字)中数据的所有位使用各自的数据线同时传输。并行接口常用于快速、近距离的设备与主机的连接。并行接口传输效率高,但设备费用也高。

常见的并行接口有打印机 Centronics 接口、IEEE-488 接口、**数模转换器**和模数转换器接口等。

(1) 打印机 Centronics 接口 打印机通用的并行接口。每次并行传送 8 位数据,最大传输距离 5 m。最大数据传输速率为 150 KB/s。采用单向传输。接口采用 36 针 D 型插头座。信号线中除了有数据线、控制线、状态线、地线和电源线外,还有一些空线供用户安排特别的用途。

EPP 接口是 Centronics 接口的改进型。它采用双向、半双工传输,最大数据传输速率可达 2 MB/s。

(2) IEEE-488 接口 也称为 HPIB 接口或 GPB 接口。它最初是 HP 公司提出的,1975 年被接受为 IEEE 标准。IEEE-488 接口采用 24 针 D 型连接器。其中有 8 根双向数据线,3 根信号交换线,5 根控制线。总线上最多可连接 15 个设备。设备分成三类:控制方、发送方和接收方。数据传送时只有一个发送方,但可有多个接收方。为使快慢不同的设备能一起工作,采用应答方式。当各方都准备就绪后,控制方通知发送方发送数据,当最慢的一个接收方接收完成后,发出应答信号,发送方发出信号通知控制方和所有的接收方,完成一次传送。

IEEE-488 接口的最大数据传输速率为 1 MB/s。最大数据传输距离为 20 m。

参考文献

1. 古辉,等. 微型计算机接口技术. 北京:科学出版社,2011

2. 袁新燕. 计算机外设与接口技术. 北京:高等教育出版社,2009 (林蒹)

shuru shuchu tongdao

输入输出通道(input/output channel) **主存储器**与外围设备之间信息传输所使用的物理通道。它能同时管理多台外围设备的工作和信息传输,以减少主机直接对外围设备的管理,实现数据处理和传输的并行工作。只作数据输入的通道称为输入通道,只作数据输出的通道称为输出通道。

由于外围设备种类繁多,传送速度和传送方式各不相同,为了提高控制效率,可采用不同的通道结构,如选择通道、成组多路通道、多路转换器通道和字节多路通道等。

选择通道 对于高速的设备,如磁盘、磁带等,要求较高的数据传输速度以成组方式与主机交换数据。对于这种高速传输,通道难以同时对多个这样的设备进行操作,只能一次对一个设备进行操作。这种通道称为选择通道。

成组多路通道 可控制多台高速外围设备以成组交叉方式分时传送数据,某台设备与主机交换数据时,允许别的设备做寻址或其他准备传输的操作,实现寻址和传送数据并行工作,以充分发挥通道高速交换数据的能力。成组多路通道在一次传输期间能传输大量的数据,是大型计算机、巨型计算机经常采用的数据传输方式。

多路转换器通道 可使计算机与若干低速外围设备(终端和打印机等)同时传输数据。多路转换器通道采用数据交叉的方法同时传输多台外围设备的数据,有时也称交换器。

字节多路通道 用于连接多个慢速的和中速的设备(行式打印机、读卡机、远程终端等),这些设备的数据传送以字节为单位。每传送一个字节要等待较长的时间,如终端设备等。因此,通道可以以字节交叉方式轮流为多个外设服务,以提高通道的利用率。多台低速设备分时占用通道和主机交换数据。一个字节多路通道可连接多个子通道,每个子通道

可连接 1 台或多台设备,不同子通道的设备可同时工作,相同子通道内 1 次只允许 1 台设备与主机交换数据。

参考文献

1. 徐炜民,严允中. 计算机系统结构. 3 版. 北京:电子工业出版社,2010

2. 孙德文. 计算机组成基础. 北京:机械工业出版社,2009

(徐良贤)

shubiao

鼠标(mouse)　一种与屏幕配合使用,可对屏幕上显示的信息进行增、删、改的直观的人机通信工具。在驱动程序的支持下,鼠标能完成图形用户接口的功能。第一个用于 IBM-PC 的鼠标是 1982 年由 Mouse system 公司研制成功的。由于它操作方便,易于使用,随着 UNIX、Windows 等带有图形用户接口的**操作系统**的普及,鼠标已成为个人计算机和工作站必备的输入设备。

鼠标因其外形(如图 1)而得名。当鼠标与计算机连通时,在显示屏上就会出现一个光标,通常是一个指向左上方的小箭头。程序所操作的区域就是箭头顶部所指的区域。当将鼠标在一个平面上移动时,其底部的传感器就把移动的方向和距离检测出来,送入计算机,控制光标作相应的运动。鼠标上带有 2 个或 3 个按钮,有的还有上网专用按钮或一个小滚轮。移动光标对准显示在屏幕上的命令或图标,按下按钮即可完成操作。

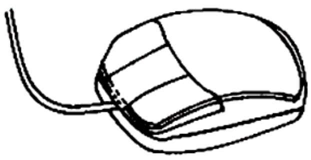


图 1　鼠标

鼠标是一种相对定位设备,不受平面上移动范围的限制。它的具体位置也和屏幕上光标绝对位置没有对应关系。当鼠标脱离平面时,屏幕上光标并不会移动,直到鼠标在平面的另一位置放下并再度移动时,光标才又会跟着运动。

鼠标的主要性能指标是动态分辨率和跟踪速度(或称辨识速度)。动态分辨率指鼠标每移动一个单位距离能检测出的脉冲数,通常用点数每英寸(dpi)表示。鼠标的动态分辨率有 200 dpi,400 dpi,800 dpi 等。跟踪速度指鼠标不发生错误的最大移动速度,2003 年已达到 10 m/s。

鼠标依照所采用的传感技术可分为机械式、光电式、机电式三种。

机械式鼠标可靠性差,现在已很少使用。光电式鼠标采用光电传感器,底部不设圆球,而是一个光电元件和光源组成的部件。当它在专用的有明暗相间的小方格的平板上运动时,光电传感器接收到反射的信号,测出移动的方向和距离。光电式鼠标分辨率高,可靠性高。但需要专用的平板,而且价贵,因此使用也不多。机电式则是上述两种结构的综合。它底部有圆球,但圆球带动的不是机械编码盘而是光学编码盘,从而避免了机械磨损问题,也不需要专用的平板。是最常用的一种。

鼠标通过电缆与主机相连接。早期个人计算机是接在串行通信口 COM 上,现在多接在 PS/2 口或 USB 口。20 世纪 90 年代以来,采用红外线、电磁波与主机通信的无线鼠标也有了广泛的应用。

(林兼)

shuxing wenfa

属性文法(attribute grammar)　允许为每个终结符和非终结符配备一些属性的文法。它既能描述程序设计语言的语法,又为其语义处理提供了方便。属性文法由 D. E. Knuth 于 1968 年引进。属性有不同的类型,可以像变量一样地被赋值。赋值规则附加于语法规则之上,赋值与语法分析同时进行,赋值过程就是语义处理过程。在推导语法树的时候,诸属性的值被计算并通过赋值规则层层传递。有的从语法规则左边向右边传,有的从右边向左边传。语法推导树最后完成时,就得到出发符号的属性值,也就是整个程序的语义。

属性有两种,一种叫继承属性,另一种叫综合属性。继承属性的计算规则为由顶向下,对语法规则来说是从左至右。综合属性的计算规则为由底向上,对语法规则来说是从右向左。例如,在

$A_p \to B_{q,r}\ C_{s,t}$　(语法规则)

$[p=q+s, r=p+t]$　(属性赋值规则)

中,p,q,r,s,t 是属性,分别属于 A,B,C 三个非终结符,其中 p 是综合属性,r 是继承属性。q,s,t 的性质要把其他语法规则中的属性赋值规则一起考虑才能确定。

规定:每个语法符号的继承属性集和综合属性集之交为空。但即使在此前提下,属性及其赋值规则仍是不能任意给定的,否则会引起问题,如属性互

相依赖,因而无法计算。D. E. Knuth 称没有循环依赖关系的属性文法为良义属性文法。然而加扎耶里等又证明了:确定一个属性文法是否良义的任何一个算法必然具有指数复杂性。因此许多人致力于寻找良义属性文法的子集。如 U. Kastens 的有序属性文法和肯尼迪等的绝对无循环属性文法。

S. Fang 在 1972 年研制的 FOLD 系统中,首先实现了属性文法,其中使用了非确定性的属性计算方法。后来出现的用确定性方法计算属性文法的系统一般都有某些限制,如对扫描的遍数和扫描的方向加以限制。此外,著名的 L 属性文法适用于由顶向下的分析,S 属性文法适用于由底向上的分析。属性文法在编译程序的机械生成中起了很大的作用,使人们不仅可以机械地生成语法分析程序,还可以机械地生成语义处理程序。 (陆汝钤)

shu

树(tree) 无回路的连通**无向图**。又称无向树(参见**图论**)。每个分支都是树的无向图称为森林。如果树 T 是无向图 G 的生成子图,则称 T 为 G 的生成树。如果森林 F 是无向图 G 的所有分支的生成树之并,则称 F 为 G 的生成森林。如果森林 F 有 n 个顶点、m 条边和 k 个分支,则 $m=n-k$。设 $\langle G,w\rangle$ 是加权图,$G'\subseteq G$,G' 中所有边的加权长度之和称为 G' 的加权长度。设 $\langle G,w\rangle$ 是加权连通无向图,G 的所有生成树中加权长度最小者,称为 $\langle G,w\rangle$ 的最小生成树。

设 G 为弱连通**有向图**,其中一个顶点的入度为 0,其余顶点的入度均为 1,则称 G 为有向树。入度为 0 的顶点称为根,出度为 0 的顶点称为叶,出度大于 0 的顶点称为分支顶点,从根至任意顶点所经过的边数称为该顶点的级,所有顶点的级的最大值称为有向树的高度。每个弱分支都是有向树的有向图称为有向森林。设 $m\in\mathbf{N}$(自然数集合),D 为有向树,a 和 b 是 D 的两个顶点,如果有一条边从 a 到 b,则称 a 是 b 的父亲,b 是 a 的儿子。如果 D 的所有顶点出度的最大值为 m,则称 D 为 m 元有向树。如果对于 m 元有向树 D 的每个顶点 v,均有 $d_0^+(v)=m$ 或 $d_0^+(v)=0$,则称 D 为完全 m 元有向树。完全二元有向树又称为二叉树。在计算机科学中二叉树特别有用,它可以用来研究算法的效率等问题。对于 n 阶二叉树 D,D 有 $(n+1)/2$ 个叶,D 的高度 h 满足:$\log_2(n+1)-1\leqslant h\leqslant(n-1)/2$。为每一级上的顶点规定了次序的有向树称为有序树。如果有向森林 F 的每个弱分支都是有序树,且也为 F 的每个弱分支规定了次序,则称 F 为有序森林。为每个分支顶点的儿子规定了位置的有序树,称为定位有序树。 (张强)

shuju beifen

数据备份(data backup) 为了防止由于操作失误、系统故障等意外原因导致的数据丢失,而将整个系统的数据或者一部分关键数据从主计算机系统的存储设备中复制到其他存储介质的过程。以保证当主计算机系统的数据不可用时,可以利用复制的数据进行恢复,避免数据的丢失。

传统上,数据备份主要采用内置或外置的**磁带机**进行备份。相对于比较昂贵的磁盘存储空间,利用磁带介质进行备份,其单位存储容量花费较低。而随着近年电子技术水平的不断提高,磁盘单位存储容量的价格也不断降低,相比以往已经大大接近于磁带单位存储容量的价格,同时由于重复数据删除技术的兴起以及磁盘相对于磁带的明显性能优势时,利用磁盘作为备份介质也越来越普遍。在构建数据备份系统时,除了需考虑备份介质的选择外,如何尽量减少备份工作对系统的额外负担、保证系统正常业务的高效运行,也是需要重点考虑的问题。备份工作往往会给系统的正常应用带来很大的负担,甚至严重影响到系统的正常运行,这样的备份工作虽然勉强可用,但不是最好的。可以说备份技术的发展,如异步复制技术、同步复制技术、快照等,也都是针对这个问题不断进行解决的过程。

考察数据备份的质量有两个关键指标:数据恢复点目标 RPO 和恢复时间目标 RTO。RPO(recovery point objective),主要描述业务系统所能承受的数据丢失量;而 RTO(recovery time objective),是指从灾难发生到重新恢复服务所需要的时间。RPO 主要针对数据丢失量,RTO 针对服务丢失时间,两者没有必然的联系,但又互相影响。备份技术的发展是不断提高这两个指标的过程。对于不同的企业、不同的应用系统、不同的高可用要求,人们对于 RPO 和 RTO 的要求也不尽相同,这需要在制订备份计划时由系统管理员来综合考虑。

常用的三种备份策略是完全备份、增量备份、差量备份。实际应用中,用户可根据需要选取其中一种或进行组合。

数据备份相比于数据容灾,在本质上有所不同。虽然数据备份和数据容灾的目的都是为了提高系统

的可用性,为了消除或者降低灾难给系统带来的影响,为了灾难过后将系统恢复到正确的状态,但是两者关心的重点不同。数据容灾的关键在于保护系统数据的在线状态,保证数据在灾难时能从容灾中心及时恢复并且无缝地向外提供数据服务,实时地保护数据。而数据备份是将某个特定时间点的系统完整、统一的数据或状态保存下来。数据备份技术并不能保证数据的实时性,一旦灾难发生,数据备份只能保证在一定时间内(比数据容灾要长得多)将数据恢复到某个时间点上的完整正确的状态。在恢复过程中,系统是不可用的;恢复完成后,数据也不能恢复到灾难发生时的正确状态,而只能是灾难之前一段时间的正确状态。

参考文献

1. 周敬利,余胜生. 网络存储原理与技术. 北京: 清华大学出版社,2005

2. Zhu B, Li K, Patterson H. Avoiding the disk bottleneck in the data domain deduplication file system. In: Proceedings of the 6th USENIX Conference on File and Storage Conference. 2008. 269-282 (谭玉娟)

shuju cangku

数据仓库(data warehouse) 面向主题的、集成的、不可更新的、随时间不断变化的数据集合,是为**决策支持系统**提供基础数据的分析型数据库。

数据仓库技术是20世纪90年代随着数据库应用由操作型处理向分析型处理扩展而兴起的计算机技术。***操作型处理***也叫**事务处理**,是指对数据库联机的日常操作,通常涉及对数据库中当前记录的查询和修改;***分析型处理***是指基于数据的分析操作和辅助决策,通常是对海量历史数据的查询和分析。前者要求响应速度快、系统的并发度高和事务吞吐量高;后者则访问的数据量非常大,查询和分析操作复杂。传统的数据库技术不能同时满足两类数据处理的要求,数据仓库技术应运而生。数据仓库本质上和**数据库**一样是长期储存在计算机内,有组织、可共享的数据集合。数据仓库和数据库主要的区别是数据仓库中的数据具有以下四个基本特征: ①数据是面向主题的; ②数据是集成的; ③数据是不可更新的; ④数据是随时间不断变化的。数据仓库中的数据组织具有不同的综合级别,一般称之为"***粒度***"。粒度越大,表示细节程度越低,综合程度越高。

数据仓库系统总体上由以下几部分组成: 数据仓库的后台工具、数据仓库服务器、联机分析处理(OLAP)服务器和前台工具,如图1所示。

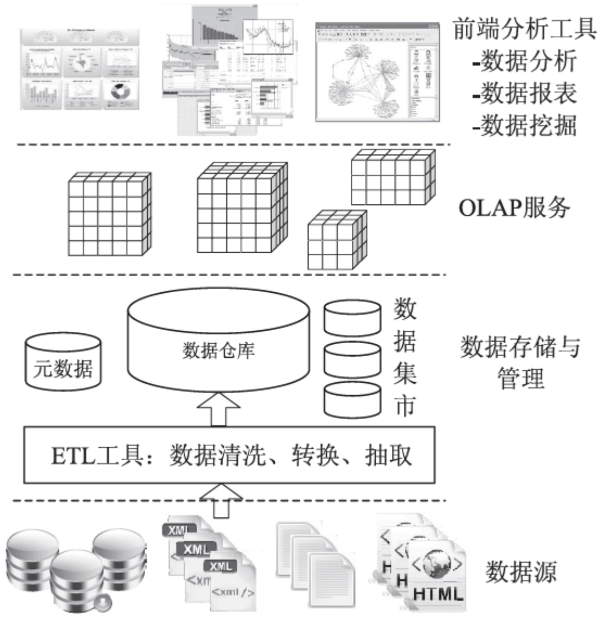


图1 数据仓库系统的体系结构

数据仓库的后台工具,包括数据抽取、清洗、转换、装载和维护工具。它们负责对各种数据源进行转换,然后装载到数据仓库中,同时维护数据仓库和数据源数据的一致性。与数据仓库有关的系统目录很大,需要在一个独立的数据库中存储与管理,这个数据库称为***元数据库***。元数据(参见**多媒体数据库**)管理工具是数据仓库系统的一个重要组成部分。

数据仓库服务器负责储存和管理数据仓库的数据,并给OLAP服务器和前台工具提供存取接口(如结构查询语言SQL查询接口)。数据仓库服务器目前一般是关系数据库管理系统(RDBMS)或扩展的RDBMS。数据集市是部门级的数据仓库。数据集市可以按业务来组织,也可以按主题或数据的地理分布来组织。

联机分析处理(OLAP)服务器为前台工具和用户提供多维数据视图(参见**联机分析处理**)。

前台工具包括查询报表工具、多维分析工具、**数据挖掘**(DM)工具和分析结果可视化工具。

在整个系统中,数据仓库居于核心地位,是数据分析和挖掘的基础; 数据仓库管理系统负责管理系统的运转,是整个系统的引擎; 而数据仓库工具则是整个系统发挥作用的关键,只有通过高效的工具,数据仓库才能真正把数据转化为信息和知识,为企业和部门创造价值发挥作用。

由于数据仓库中的数据是不可更新的,数据仓库管理系统的事务处理功能可以简化。另一方面,由于数据量大,数据仓库对聚集查询处理的要求较高。如何提高查询处理性能是数据仓库技术面临的一大挑战。TPC-H、TPC-DS 是对数据库中大量数据复杂查询的基准性能测试,可以用它们来衡量和比较数据仓库的性能。

数据仓库系统是一种解决方案,不是可以买到的现成产品。建立数据仓库系统是一项长期、艰巨和有风险的信息系统工程。目前,数据仓库技术、联机分析处理技术和数据挖掘技术已经成为企业商务智能的三大技术支柱,并在金融、保险、税务、零售、航空及医疗等行业获得广泛应用。

参考文献

1. Han J W, Kamber M, Data mining: concepts and techniques. 3rd ed. Morgan Kaufmann, 2011

2. 王珊,李翠平,等. 数据仓库与数据分析教程. 北京:高等教育出版社,2012

3. Inmon W H. Building the data warehouse. 4th ed. John Wiley & Sons Inc. , 2005

(王珊　陈红　张延松)

shuju chuli sulü

数据处理速率(processing data rate,PDR) 美国及巴黎统筹委员会用来限制计算机产品出口的系统性能指标估算方法,它按如下的公式来定义:

$$PDR = \frac{L}{R}$$

式中,L——指令平均长度,即每条指令传送数据的平均位数,b;

R——指令的平均执行时间,μs。

$$L = 0.85G + 0.15H + 0.4J + 0.15K$$

$$R = 0.85M + 0.09N + 0.06P$$

式中,G——定点指令字长(≥24b),b;

H——浮点指令字长(≥30b),b;

J——定点数长度,b;

K——浮点数长度,b;

M——定点加法时间,μs;

N——浮点加法时间,μs;

P——浮点乘法时间,μs。

参加运算的两个数之一应在累加器或用作累加器的存储单元中,另一个在内存中,运算结果留在累加器或用作累加器的内存单元中。

从上面的公式可见,PDR 是根据定点、浮点加和乘法指令执行时间以及它们相关的数据宽度,按一定的混合比例对中央处理机与主存的数据处理速率进行计算的。它允许并行处理及先行取指令的功能,这时,所取的是平均指令执行时间。

带有高速缓存的计算机,因为存取速度加快其 PDR 值也就相应提高。

PDR 已在 1991 年 9 月停止使用,改为用 CTP(综合理论性能)计算。

参考文献

陈兴业. 计算机系统性能评价. 广州:华南工学院出版社,1987

(吴荣泉)

shuju chuanshu

数据传输(data transmission) 指数据发送端(sender)依据与接收端约定的规程或协议(protocol),将发送的数据进行编码后,通过传输介质(medium)上的传输信道(或称链路)传输到数据接收端(receiver)的过程和技术。

早期的数据传输主要指电报等各种文本类消息的交换,现代的数据传输主要指计算机数据和多媒体信息的交换。

由于历史的原因,数据一般以字节(Byte,8 位二进制数)为基本单元表示;以字节为基本单元的数据传输通常转换为以位(Bit,一位二进制数)为单元的数字传输,此时,在通信介质上的传输信道称为数字传输信道(也称二进制传输信道),传输速率采用 bit/s 为单位。采用数字传输的主要优点在于,数据传输可以采用统一的数字传输接口去适应各种通信介质和各种通信技术。

数据传输也可以进一步分为传统数据传输和广义数据传输两种类型。传统数据传输主要考虑点到点的传输信道(链路),通常采用以下几种传输和控制模式:并行与串行传输技术,异步与同步传输技术,单工与双工传输技术等。

广义数据传输主要考虑面向网络的数据传输,其主要技术与数字通信网络领域的相关内容交叉,如传统的综合业务数字网络(ISDN)技术、宽带综合业务数字网络(B-ISDN)技术、异步传输模式(ATM)技术等,而目前使用最为广泛的是计算机数据网络——互联网。

并行传输与串行传输

并行传输(parallel transmission)指数据在多条并行的数字传输信道上同时进行传输的方式。例如,台式计算机中央处理器(CPU)的内部数据总线

从早期的8位、16位、32位逐步发展到64位等,拥有64位数据总线宽度的CPU可以在一个传输周期内向内存同时传输8个字节(64位二进制)的数据。广义的情况下,光纤波分复用传输和频分复用传输也属于并行传输的方式。并行传输使用的信道多,一般用于大容量通信的环境。

串行传输(serial transmission)指数据转换为二进制数,使用一条数字传输信道按时间顺序逐位传输的方式。该方式仅需要一条数字传输信道,通常较为节省传输介质,如果需要高速传输,则需要增加传输的速度。由于串行传输需要的传输介质最少,连接相对简单,通常用于计算机系统外设间的有线通信,例如USB接口技术采用了超高速串行通信方式,广泛用于计算机与外设、计算机与手机等设备之间的通信。

异步传输与同步传输

同步传输(synchronous transmission)指数据的发送端与数据的接收端时钟完全同步,发送端的数据转换为二进制比特流,连续发送给接收端,根据同步的时钟,接收端判定接收比特流中的数据起始位置,将二进制比特流恢复成数据。传统的数字通信系统主要来源于支持数字化语音的通信系统,在支持连续话音通信时,采用同步传输技术的效率极高,在这样传统的数字传输通道上,数据传输也具有高效率的特点,例如计算机互联网通常采用IP over SDH同步数字系统进行数据传输。但是,同步传输需要接收端与发送端的时钟保持完全同步或采用附加信道专门传输时钟,因此增加了系统的复杂性。

异步传输(asynchronous transmission)指数据的发送端在传输的二进制比特流前后添加特殊的起止标志,接收端通过识别起止标志,判定接收比特流中的数据起始位置,将二进制比特流恢复成数据。例如计算机键盘与主机的通信,使用者按下一个字母键、数字键或特殊字符键时,将产生一个8比特的ASCII字符代码数据,键盘使用二进制发送时,前面加一个起信号“0”,而在发完后加一个止信号“1”。接收端通过检测起、止信号,即可分离出所传输的字符代码。采用异步传输技术时,数据的同步定时非常简单,由于每一个数据都加上了自己的“同步”信息,因此收、发两端的时钟误差不会积累,不需要精确同步。缺点在于传输每一个数据需要额外的“同步”开销。在上面的例子中,每传输8个比特的ASCII字符代码,需要传送两个比特的起止信息,额外开销占总传输量的20%。

参考文献

1. 胡道元. 计算机网络(高级). 北京: 清华大学出版社, 1999

2. Stallings W. Data and computer communications. Englewood Cliffs, NJ: Prentice Hall, 1997

(张凌)

shuju chuanshu moshi

数据传输模式(data transfer mode)　指一组数据(数据包)从发送端传输(传递)到接收端的模式,也称数据包传递模式。这里的数据包通常指一组计算机数据。传递模式主要指发送端和接收端所选择的同步方式。常用的模式有同步传输模式STM(synchronous transfer mode)和异步传输模式ATM(asynchronous transfer mode)两种。

同步传输模式

同步传输模式(synchronous transfer mode, STM)的使用方法:在数据包传输开始之前,发送端和接收端需要确定传输开始的时间、速率、时钟的同步、数据包的大小、传输过程的差错控制等参数,然后,双方同时共同完成数据包传输。同步传输模式的重点在于双方必须同时完成数据包传输,该技术通常用于将计算机数据包转换到SONET/SDH数字同步通信网上传输时使用。

异步传输模式

异步传输模式(asynchronous transfer mode, ATM)的使用方法:发送端将数据包分割成为固定长度的数据分组,加上虚拟电路标识、流量控制、检错等信息,形成信元(cell)后,通过数字传输信道或ATM网络直接进行传输,接收端将接收到的信元组装并恢复成数据包。ATM技术是在“存储-转发”技术、分组交换技术基础上发展起来的一种传输模式,在这一模式中,数据源的数据包被组织成固定长度为53个字节的信元,随时发送至ATM网络系统,ATM系统采用信令控制的动态复用与交换节点完成信元的交换,送达接收端。在ATM系统中,数据源的数据不需要定时(或同步)发送,到达的时间也不完全与发送同步,因此将这种传输模式称为异步传输模式。

参考文献

1. 胡道元. 计算机网络(高级). 北京: 清华大学出版社, 1999

2. Stallings W. Data and computer communica-

tions. Englewood Cliffs, NJ: Prentice Hall, 1997

（张凌）

shuju chuansong

数据传送（data transfer） 数据在计算机内部或计算机之间或计算机系统与其他系统之间的传送操作。

数据传送可以分为3个层次：①数据从一个计算机系统传送到另一个计算机系统；②在计算机系统中，数据从一个外围设备（或存储单元）传送到另一个外围设备（或存储单元）；③数据在中央处理器内部各寄存器之间的传送。数据在传送的过程中，不应改变其格式和内容。

衡量数据传送的性能指标主要有：①数据传送速度 在数据通道上传送数据的速度，称为***数据传送率***，也称数据传输速率，一般以每秒传送的字节、千字节或兆字节的数量作为度量的单位，记为B/s，KB/s或MB/s。在有的情况下，也用每秒传送二进制信息位的数量作度量单位，记为b/s，Kb/s或Mb/s。有时也用***平均数据传送率***（或称***有效数据传送率***）来度量，这是指在一段时间内，由一个外围设备（或存储单元）到另一个外围设备（或存储单元）传送数据的速率。这段时间包括数据块、字或记录之间的间隔时间，但不包括程序启动、查找和停止等所用的时间。②***数据宽度*** 指每次并行传送的数据的位数，例如8位，16位，32位，64位。

数据传送有多种不同的分类方法。按数据传送的方式，可分为：①并行传送 在并行传送通路上，输出数据的所有信息位并行地同时被传送到目的设备。在总线的工作频率相同的条件下，并行传送的速度比串行传送快，但所用的器件较多。②串行传送 在串行传送的通路上，输出数据中的各信息位依次一位一位地传送。同样在总线的工作频率相同的条件下，串行传送的速度比并行传送慢，但逻辑实现较简单，所用的器件较少。

按传送双方是否采用统一的时钟，数据传送可分为：①***同步数据传送*** 传送双方采用统一的时钟脉冲。同步传送的优点是传输速率高，缺点是受干扰后易引起同步错。②***异步数据传送*** 传送双方不采用统一的时钟脉冲，发送和接收设备各按自己的时钟频率工作。

按数据传送的控制方法，可分为：①无条件传送 不查询接收设备是否处于就绪状态而强迫执行操作。为使传送可靠，程序员必须了解接收设备的动态工作情况。②条件传送 满足条件时才进行的数据传送。为此，控制程序必须设有测试接收设备工作状态的指令。③程序中断传送 由程序控制，采用中断方式进行的数据传送。当程序运行到某一时刻，由程序控制启动指定的外围设备，然后主机继续执行原来程序。外围设备做好数据传送准备后，向中央处理机发出中断请求信号，中央处理机经过中断处理后，停止正在运行的程序，转向执行传送指令，以完成主机和外围设备之间的数据传送，传送完毕后，再返回执行被中断的程序。④**直接存储器存取**（DMA） 外围设备与存储器之间不需要中央处理机干预而直接由DMA控制器来控制的高速数据传送。

参考文献

1. 孙德文. 微型计算机技术. 3版. 北京：高等教育出版社，2010

2. 徐炜民，严允中. 计算机系统结构. 3版. 北京：电子工业出版社，2010

（孙德文）

shuju dianlu shebei

数据电路设备（data circuit-terminating equipment，DCE） 指位于**数据终端设备** DTE 和传输线路（data transmission circuit）之间的**数据通信设备**。数据电路设备DCE一方面向各种不同的数据终端设备DTE提供标准接口的数据传输服务，另一方面通过调制解调或复用交换等技术将数据送入通信传输介质或数据传输网络。

数据电路设备DCE与数据终端设备DTE之间的接口称为**数据通信接口**，常用的接口标准有RS232C、RS422、RS449、V.24、V.35、X.21等。

数据电路设备DCE可以采用点到点，或者一点到多点的传输信道或传输网络。传输信道的传输介质，可以是电话铜线、有线电视电缆、光纤、无线信道等，对于不同的传输介质，需要采用不同的调制解调技术，此时数据电路设备通常称为电话调制解调器（modem）、无线调制解调器（也称无线猫）、电缆调制解调器（cable modem）和光纤调制解调器等。

当两个数据终端设备DTE需要直接相连时，可以采用零调制解调器（null modem）或交叉电缆互相连接。

参考文献

1. Forouzan B A. 数据通信与网络. 2版. 吴时霖，周正康，吴永辉，等译. 北京：机械工业出版社，2002

2. Gilbert Held. 数据通信. 6 版. 戴志涛, 卞佳丽, 郑岩,译. 北京: 人民邮电出版社, 2000

(王昊翔　张凌)

shuju jicheng

数据集成(data integration)　把不同来源、格式、特点和性质的数据在逻辑上或物理上有机地组合为一个整体,从而为各数据拥有者提供全面的数据共享的技术。其中,**数据共享**(data sharing)是指为提高数据资源的利用率,由多个用户(程序)按一定规则共同享用数据库中数据的一种技术。

数据共享是数据库和文件系统的重要区别之一。在数据库出现以前,数据处理以程序为中心,每一程序都有各自所用的数据文件。同一数据为多个程序使用时要重复存放多份,而数据库的数据为有关用户公用,集中管理所需的全部数据。数据库的数据共享有下列好处:①减少重复存放,节省存储器资源,②简化了对共享数据的修改;③保证了数据一致性。

为了有效的解决数据共享问题,数据集成问题在 20 世纪 80 年代被首次提出。在 20 世纪 90 年代,随着网络技术的发展,大量在线数据源为企业数据管理带来了新契机,同时也推进了数据集成的研究,促成了商业化的数据集成系统的出现。

具有代表性的数据集成方案有 ETL(extraction-transformation-loading)和 I3(intelligent information integration)解决方案。

ETL 的中文名称为数据提取、转换和加载。ETL 负责将分布的异构数据源中的数据,如关系数据、无格式数据文件(flat data file)等,抽取到临时中间层后进行清洗、转换、集成,最后加载到数据仓库或数据集市中,成为联机分析处理、数据挖掘的基础。常见的 ETL 工具有:OWB (oracle warehouse builder)、ODI (oracle data integrator)、Informatic PowerCenter、AICloudETL、DataStage、Repository Explorer、Beeload、Kettle、DataSpider 等。

I3 解决方案中的数据集成系统的结构如下:最底层是异构数据源,其上是包装器(wapper)程序,用于将查询下发给数据源,并接收从数据源返回的结果,再将结果转化为恰当的格式传给上层结构。用户使用顶层的中间模式(mediated schema)编写查询,该模式通常与应用相关。建立数据集成系统的关键在于数据源描述。数据源描述给出了各个异构数据源的性质,其关键组成被称为**语义映射**(semantic mappings),语义映射将数据源的模式准确地映射到中间模式。

在数据集成的过程中,主要需要解决好模式匹配、语义集成、查询分解等问题。

(1) 模式匹配是指将两个不同的数据库模式作为输入,计算模式元素之间语义上的对应关系的过程。由于最底层的数据源系统多是独立开发且相当自治的,描述数据的数据模型或存储结构不可能完全相同,因此需要将不同的数据库模式相互匹配,从而能够提供给用户一个统一的中间模式,使其能够方便快捷地编写查询。

(2) 语义集成以本体(ontology)为关键支撑工具,目的是它使得数据在表示模式异构和隐含语义不同时,也能够实现共享和交换。本体是对共享概念明确的形式化表示,它使结构化语义描述成为可能,与数据库模式一样,它可以表示元数据,却比数据库模式的表达能力更强,更易于形式化。

(3) 查询分解首先将用户发出的查询分解、重写为针对每个数据源的、与原查询等价的子查询,然后经过包装器转换为对底层数据源支持的数据查询。

(4) 在数据集成系统中,无论数据被存储于何地,都很难保证所有数据完整、一致、准确和最新,因此数据质量问题也是数据集成领域中亟待解决的问题之一。

当前,数据库集成系统的主要应用有企业数据管理、互联网数据管理、大规模科学数据管理以及政府部门合作等方面。其中,在企业数据集成领域,已经有很多成熟的框架可以利用,如联邦数据库系统、中间件模式和数据仓库等。

参考文献

Ullman J D. Principles of database systems. 2nd ed. London: Pitman Publishing Ltd., 1987

(李建中　楼荣生　朱扬勇　邹兆年)

shuju jiegou

数据结构(data structures)　由若干数据成分按照一定方式构成的复合数据以及作用于其上的函数或运算。数据成分及其间的数据约束关系合称为数据结构的逻辑构成或逻辑结构。数据结构从数学上可以用适当的数学结构以及在其上的函数变换统一地定义。

今以字符串为例,它的逻辑结构是一组同一类型(字符型)的字符,以及在该字符串中字符间的前

后顺序关系。在此结构上的常见运算有:求字符串的长度、将字符串分为两部分(首字符及其余部分)以及求两字符串拼接后的新串等。在设计数据结构阶段,应该视应用需要确定适当的运算。这对于设计字符串数据结构是很重要的。

一个数据结构的存储结构是指它在计算机中所需用的存储空间、空间的构成结构及对该存储结构的访问方式等的统称。一般来说,存储结构的具体设计是在明确该数据结构的定义之后进行的。在传统的程序设计语言(如 PASCAL,C 等)中,定义数据结构的基本途径是采用数据类型。简单的数据结构是用单一的标准数据类型,如整型、字符型、布尔型或实型等所定义的。在使用时,程序对其采取整体性访问方式,即读写访问只对整体而不涉及对其某构成部分的访问(如其中某二进位的内容)。由若干较简单的数据结构,运用程序语言所提供的复合构成方法,诸如数组、记录、集合等,可以构造更为复杂的数据结构。复杂数据结构的特点是:对它既可进行整体性访问,也可单独对某构成成分进行访问。

以上所述,虽已被多数熟知的程序设计语言所采用,但是有其局限性。它偏重于逻辑结构及存储结构的构成方面,而对其运算等语义部分的描述不足。根据抽象数据类型理论,程序应将数据结构的逻辑构成和它的运算在一起定义,并封装成一个整体。而且数据结构的具体实现,包括采用的存储结构和运算操作的具体算法实现,应该与上述封装相分离,分别加以描述。近年来被广泛使用的 C ++、Java 等面向对象语言(以及早年的 Ada、Smalltalk 等)皆采用了这一思想。抽象数据类型将逻辑设计和物理实现的分离,使得人们可以在适当的抽象层次上考虑程序的结构和算法,并很好地支持了软件复用。存储结构涉及存储分配和数据访问两个方面。鉴于计算机的主存储器具有随机访问及相邻地址顺序存储等物理特点,常把经常被同时访问的数据安排在相邻的物理存储区域内,这称为顺序存储结构。一维数组、字符串和记录等定长的数据结构多采取这种结构,并运用静态存储分配方式:即在程序编译连接时(程序运行前)确定所占用的存储区域的物理地址范围。对顺序存储结构而言,常用的数据访问方式有顺序访问、索引访问及散列访问等多种。另一种存储结构是链接结构。与顺序存储结构不同,它常常采用动态存储分配方式:在程序运行中当需要创建新的数据结点时刻,申请和分配存储空间;当删除数据结点时则及时释放其占用空间。这种分配方式适合用于变长度的数据结构,如具有递归逻辑关系的结构,如表、树等情形。

数据的逻辑结构从比较抽象的角度——与存储结构相对独立,刻画数据结构所具有的数学性质:将数据元素抽象为结点,数据元素间的关系当作连接结点的边,而访问数据结构的运算则描述为离散数学结构上的运算。常见的逻辑结构有线性结构、树结构、图结构等。

数据结构与算法呈相互依赖的关系。只有恰当地确立了问题的结构,才能选择和设计合适的解决方法。数据结构与算法的良好设计是有效使用计算机的基本前提。

参考文献

1. Wirth N. Algorithms + Data Structures = Programs. London: PrenticeHall Inc., 1976

2. 张铭,王腾蛟,赵海燕. 数据结构与算法. 北京:高等教育出版社,2008

3. 中文 Wiki“数据结构”词条 http://zh.wikipedia.org/wiki/数据结构

(许卓群)

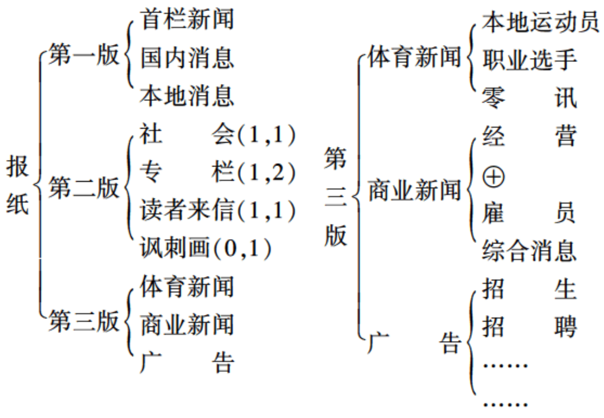
shuju jiegouhua xitong kaifa fangfa

数据结构化系统开发方法(data structured system development method) 一种面向数据结构的软件开发方法。又称 **Warnier-Orr 方法**。该方法利用 Warnier 图表示数据的层次结构,并且依据一定的步骤把层次的数据结构映射到程序结构。

1974 年法国计算机科学家 J. D. Warnier 提出了一种表示信息层次结构的图形工具——Warnier 图,他利用三种结构成分,即顺序、选择、重复来表示信息的层次结构,进而导出软件的结构。80 年代初,K. Orr 将其扩充,形成了数据结构化系统开发(DSSD)方法。该方法既考虑了数据流和功能特性,也考虑了数据的层次结构。

Warnier 图是利用树形结构表示信息层次结构的一种紧凑而直观的图形工具。用 Warnier 图可以表明信息的逻辑组织,它可以指出一类信息是重复出现的,也可以表示特定信息在某一类信息中是有条件地出现的。因为重复和条件约束是说明软件处理过程的基础,所以很容易把 Warnier 图转变成软件设计的工具。

图 1 以报纸自动编辑系统为例,说明这种图形工具的用法。图(a) 是报纸各版的构成,图(b) 则是第三版的详细栏目。花括号表示层次关系,括号



(a) Warnier 图　　　　(b) 第三版细目图

图 1　报纸自动编辑系统 Warnier 图

内从上到下是顺序的信息项;异或符号"⊕"表明一类信息或一个数据元素在一定条件下才出现,而且在这个符号上、下方的两个名字只能出现一个;一个名字右边的圆括号中的符号指明了这个信息类(或元素)在这个数据结构中重复出现的次数。

数据结构化系统开发方法把系统开发分为分析和设计两个阶段。

在分析过程中,首先研究应用环境,即从信息的产生者和接收者的观点来观察数据是如何在两者之间流动的。然后,用类似 Warnier 图的层次结构表示方法来表达应用的功能。最后,再用 Warnier 图来描述应用结果。

这一方法在需求分析时会涉及信息域的所有属性:信息流、信息内容和信息结构,其分析步骤包括:

(1) 确定应用环境　为了确定问题的应用环境,对问题作出描述,需要回答以下三个问题:要处理的信息项有哪些?谁是信息的产生者和接收者?每个信息的产生者或接收者都与环境中的哪些信息项有关?DSSD 方法以实体图作为工具回答这些问题。

(2) 表达应用功能　研究跨越系统边界的信息流可以弄清楚所开发系统应该实现的功能。使用类似 Warnier 图的层次结构表示法,可以把信息和加工联系起来,称之为作业线图。在该图中,每个信息流项是前面编号信息项与产生所需项的过程结合起来得到的。每个加工过程由一个处理说明细化。该说明包括输出、动作、动作的频率及输入。

(3) 给出应用结果　DSSD 方法要求对系统的输出建立雏型,以表明主要的系统输出及构成输出的信息项组织。有了这一雏型就可利用 Warnier-Orr 图来模拟信息的层次结构。

DSSD 方法设计过程以分析过程的需求规约作为设计的根据,需求规约包括应用环境、功能描述及应用结果。所有表述这些信息的数据和图表是 DSSD 设计的基础。设计过程着眼于系统输出、界面设计和软件的过程设计,它以输出数据结构(称逻辑输出结构)和对应过程(称逻辑处理结构)的详细描述作为设计过程的结果。

DSSD 是一种公式化和综合性的设计方法,其设计步骤为两部分:

(1) 逻辑输出结构(LOS)的获得　LOS 的获得可以通过以下 4 个步骤:①对问题进行描述并对有关信息进行量化,将所有可区别的不可再分的数据项(称为原子数据项)一一列出。②指明各原子数据项出现的频率。③对可被再分的数据项进行量化。④作出 LOS 的图表表示。

逻辑输出结构是信息项的分级表示,其中的信息项由基于计算机的系统输出构成。提取 LOS 的第一步是分离所有的原子数据项,这可由对问题陈述的了解,或审查映象报告的格式来取得。其次,各原子数据的出现频率应明确地指出。当定义了所有的原子数据项并列出其出现频率后,设计者再查出具有全局性质的信息项,即所谓全局信息项。

(2) 逻辑处理结构(LPS)的提取　逻辑处理结构(LPS)处理对应 LOS 所需软件的过程表示。DSSD 方法中提取 LPS 与早期 Warnier 方法提取程序逻辑构造(LCP)有相似之处。每一个全局数据项都成为不断加入过程指令的重复结构。提取 LPS 的步骤如下:①从表示 LOS 的 Warnier-Orr 图中提出所有原子数据项。②将 BEGIN 和 END 作为分界符加入所有全局数据项。③定义所有初始和结束的指令或进程。④指出所有可计算的和非数字的进程。⑤指明所有输出指令和进程。⑥指明所有输入指令和进程。

DSSD 方法是一种系统的软件分析和设计方法。由于这种开发方法提供了具体的工作步骤以取得详细的过程设计,所以为自动生成源代码提供了可能。由此方法形成的自动代码生成系统可通过对逻辑和实体输出及过程结构的精确描述来生成所需代码。

参考文献

1. Warnier J D. Logical construction of programs. Van Nostrand Reinhold,1974

2. Orr K T. Structured systems development. New

York：Yourdon Press，1977 （郝克刚　葛玮）

shujuku

数据库（database）　长期储存在计算机内、有组织、可共享的大量数据的集合。数据库中的数据按一定的数据模型组织、描述和储存，具有较小的冗余度、较高的数据独立性和易扩展性，并可为各种用户共享。整个数据库在建立、运行和维护时由**数据库管理系统**（DBMS）统一管理、统一控制。用户能方便地定义数据和操纵数据，DBMS能保证数据的安全性、完整性，有效地提供多用户对数据并发使用时的并发控制以及发生故障后的数据库恢复。数据库是**数据库系统**的一个重要组成部分。

数据库在历史上曾出现过三个词：databank，data base和database。

databank出现在20世纪60年代后期的一个数据系统中。当时，databank储存的主要是数字和符号，并通常用于股票、债券等领域。以后，databank一词逐渐被database所取代。data base出现在20世纪60年代初，顾名思义是存放大量数据的“仓库”。database出现在1964年数据处理方面的文献，最早由美国军方人事部门采用。20世纪60年代后期database一词为整个数据处理领域广泛采用。现在，data base和database可以混用，而database的使用更为普遍。

数据库中的数据有逻辑和物理两个侧面。数据逻辑结构的描述称为***逻辑模式***，逻辑模式分为描述全局逻辑结构的全局模式（简称模式）和描述具体应用的局部逻辑结构的子模式（外模式）。数据物理结构的描述称为***存储模式***（内模式）。这三个模式是对数据的三个级别的抽象。数据库同时提供了子模式与模式之间、模式与储存模式之间的映射，从而保证了数据库中的数据具有较高的物理独立性和一定的逻辑独立性。

数据库不仅储存实体，而且体现实体之间的联系。数据库中的数据包括：数据本身、数据描述（即对数据外模式、模式、内模式的描述）、数据之间的联系和数据的存取路径。数据库中的数据是整体结构化的，数据不再面向某一程序，从而大大减小了数据冗余度和数据之间的不一致性。同时，对数据库的应用可以建立在整体数据的不同子集上，使系统易于扩充。

人们借助计算机和数据库技术科学地存储和管理大量的、复杂的各种类型的数据，以便能方便而充分地共享和利用这些宝贵的信息资源。

当前，数据库中存储的数据类型由传统的数字、字符扩展到文本、声音、图形、图像、XML、HTML等各种类型。应用领域从传统的事务处理扩展到分析处理及科技、经济、社会、生活的各个领域。

参考文献

1. 王珊，萨师煊. 数据库系统概论. 4版. 北京：高等教育出版社，2006

2. Date C J. An introduction to database system. 8th ed. Reading：Addison-Wesley Publishing Company，2005 （王珊　陈红）

shujuku anquan

数据库安全（database security）　数据库安全是在两个层面上为数据库的数据所提供的一种保护。第一个层面是当对数据库进行有损安全的人为的非法操作时，它能保证数据免受未经授权的访问、有意和无意的破坏，从而避免由此造成的数据丢失、篡改以及数据的不一致。在这个层面上**数据库管理系统**所提供的数据安全措施主要包括***访问授权***、***数据审计***、***数据加密***等方法。第二个层面是当系统的硬件或软件出现不可避免的故障或事故而导致数据的损害时，它仍然能够提供正确的数据，保证数据的可用性。在这个层面上数据库管理系统所提供的数据安全措施主要包括***数据复制***、**数据备份**等措施。

访问授权（access authorization）　针对数据库中不同部分的数据为用户提供不同的访问权限，它包括读权、插入权、更新权、删除权、索引权、资源权、变动权、废弃权等。读权只允许用户读取，但不能修改数据；插入权允许用户插入新的数据，但不能修改已有数据；更新权允许修改数据，但不能删除数据；删除权允许用户删除数据；索引权允许为数据建立或删除索引；资源权允许用户在数据库里建立新的关系；变动权允许在关系表中增加或删除属性；废弃权允许用户从数据库中删除指定的关系。已经获得数据授权的用户可以将这种授权授予其他用户。

数据审计（data auditing）　采用日志的手段记录对数据的插入、删除、更新的操作，同时记录实行操作的用户、时间等信息。因此，通过跟踪和分析这样的日志，可以找出违反数据安全的行为和个人，尽管这是一种被动的安全防范措施。

数据加密（data encryption）　对较为敏感的数据所提供的一种更加严密的安全措施。在这种情况下，数据以加密后的形式存放在数据库里，只有获得

密钥的用户才能获得加密数据的原有形式。通过不同的密钥和密钥管理，可以进一步加强对加密数据的保护，从而提高数据安全性。

为了保证第一个层面上数据库的安全，仅仅在数据库内部采用前面所说的措施还是不够的，还应当在数据库系统赖以运行的操作系统和网络环境上提供足够的安全保证，避免攻击者绕过数据库系统，利用操作系统或网络中存在的安全漏洞窃取数据或破坏数据库。另外，支持数据库系统的整个计算机系统也应采用防侵入、防电磁泄漏等物理措施以加强安全。对于授权的用户应进行必要的审查和管理以杜绝影响数据库安全的内部隐患。要想万无一失地保证数据库安全，使之免于遭到任何蓄意的破坏几乎是不可能的。但高度的安全措施将使蓄意的攻击者付出高昂的代价，从而迫使攻击者不得不放弃他们的破坏企图。

数据复制（data duplication） 把同样的数据复制到不同的硬盘或机器上并保持数据更新的一致性。这样当一台硬盘或机器发生故障或事故时，系统可以及时地从其他硬盘或机器上获得同样的数据，使得应用系统不受影响。

数据备份（data backup） 将数据定期地转储到另外的存储设备上。当出现故障或事故时，可以利用这些备份的数据以及日志，恢复数据，继续数据库的正常运行。

参考文献

1. 王珊，萨师煊. 数据库系统概论. 4 版. 北京：高等教育出版社，2006

2. Silberschatz A, et al. Database system concepts. 4th ed. McGraw-Hill Inc., 2002

（周立柱）

shujuku bingfa kongzhi

数据库并发控制（concurrency control in database systems） **数据库系统**中对访问同一数据库的多个事务进行合理调度，使得这些事务同时运行时不破坏相应数据库的数据完整性的一种机制和过程。

当多个事务并发执行时，如果使用不正确的并发控制机制，即使每个事务单独执行时均能保证事务的正确性和数据库状态的一致性，由于多个事务之间的相互影响也可能会出现意想不到的错误结果，例如：丢失更新、读脏数据、求和结果错误等。

并发控制的目的是要避免事务之间的相互影响，保证事务的隔离性，即让并发中的每个事务感觉不到其他事务也在并发执行。对参加运行的全部事务中的操作的顺序进行合理的安排，称为**调度**。如果一种调度能保证多个事务的并发执行结果与这些事务按照某个顺序顺次执行的结果是一样的，则可避免多个事务之间的相互影响，得到正确的执行结果，我们称这种调度具有**可串行性（serializability）**。具有可串行性的调度能保证事务执行的正确性和数据库状态的一致性，**可串行性**是判定调度正确性的准则。

并发控制机制保证事务调度满足可串行性，采用的主要方法有：封锁方法、时间戳方法和乐观方法。

封锁方法（locking scheduler） 基本思想是，事务访问某数据对象前要对该数据对象加锁，如果该数据对象已被加锁，则要等待该锁被打开以后才能加锁，也就是说，在一个时刻只允许一个事务对某一数据对象进行访问操作。如果一个事务的所有加锁操作都放在第一个开锁操作之前，即对每个事务的调度均明显分为加锁和开锁两个阶段，则称该调度遵守两段封锁（two phase locking，2PL）协议。遵守 2PL 协议的并发控制可保证对事务的调度满足可串行性，事务执行结果的正确性以及数据库状态的一致性。

大多数并发控制算法是基于封锁的，基于封锁的并发控制算法的副作用是可能引起死锁。**死锁**（deadlock）指的是多个事务竞争共享的数据对象而处于永远等待的状态，几个事务竞争共享的数据对象又被称为发生冲突。例如事务 T1 已对数据对象 D1 加锁而等待事务 T2 打开对数据对象 D2 加的锁，事务 T2 已对 D2 加锁而等待事务 T1 打开 D1 的锁，于是这两个发生冲突的事务都处于永远的等待状态而不能执行下去。在数据库系统中检查和处理死锁是一件很复杂的事情，时间戳和乐观方法可避免死锁的产生。

时间戳方法（timestamp ordering） 给每一个事务盖上时间戳（时间戳标记的是开始执行事务的时间，每个事务具有唯一的时间戳），并按照这个时间戳来解决事务的冲突操作。如果发生冲突操作，则回退时间戳标记的时间较晚的事务来保证其他事务的正常执行，被回退的事务被赋予新的时间戳并从头开始执行。

乐观方法（optimistic scheduler） 认为事务执行时很少发生冲突，因此不对事务进行特殊的管制，而是让它自由的执行，事务提交前再进行正确性检查。

如果检查后发现该事务执行中出现过冲突并影响了可串行性,则拒绝提交并回退该事务。乐观方法又被称为**验证方法(certifier)**。

参考文献

1. Bernstein P A, Hadzilacos V, Goodman N. Concurrency control and recovery in database systems. Addison Wesley Publishing Company, 1987

2. 周龙骧. 数据库管理系统实现技术. 武汉:中国地质大学出版社,1990

(周龙骧　王翰虎　杨冬青)

shujuku guzhang huifu

数据库故障恢复(fault recovery in database systems)　将数据库恢复到故障发生前的一致状态的机制,它是一种保证数据库可靠性和容错性的重要机制。

故障可分为事务级故障、系统级故障和介质级故障。

事务级故障是指单个事务发生的故障,即由于事务内部或外部原因造成某个事务无法继续正常执行。例如事务中的某个操作失败、事务申请资源过多、事务执行的结果是错误的等。这类故障的影响范围局限在该事务本身。

系统级故障是指硬件或系统软件故障造成系统运行终止。例如操作系统崩溃、数据库系统发生故障、停电事故等。系统故障局限于系统范围,影响多个甚至全部事务的执行。

介质级故障是指保存数据的存储介质发生了故障,造成数据库中数据的丢失或损坏。

恢复的基本思路是采用"冗余"技术,即在数据库系统正常运行的过程中,使用日志、检查点、保存点或转储技术储存恢复时可能需要的数据信息,当发生的故障引起数据库的数据被丢失或损坏时,数据库系统中的恢复机制使用这些冗余存储的数据将数据库恢复到故障发生前的一致状态。

日志(log)　记录数据库事务对数据库的操作历史及相关数据,包括数据库系统中正运行的事务状态及其更新活动,例如事务的开始、提交或中止,某事务对某数据元素进行修改的改前旧值和改后新值等。故障发生后根据日志中记录的内容,重做(redo)已经执行过的事务对数据库的操作或撤销(undo)已经执行过的事务对数据库的修改,将数据库恢复到故障发生前的一致状态。写日志文件必须遵循先写日志的原则,即任何数据对象修改后的值被写到磁盘之前,必须先把相应的日志记录写到磁盘中。

检查点(checkpoint)技术　指按一定的时间间隔强制地刷新日志和数据缓冲区,建立一个准一致的数据库状态。发生故障后,用故障发生前最近的一个检查点开始记录的日志和数据来恢复数据库,避免从头开始进行恢复的问题,改善恢复的效率。检查点由数据库系统中的恢复机制设定。

保存点(savepoint)　一种由用户在应用程序中声明用来回退部分事务(保存点之后的那部分)的机制。系统在保存点处强制地刷新日志缓冲区和数据缓冲区,发生故障后从故障发生前最近的一个保存点处开始恢复数据库,将事务回退到用户声明的保存点处。

数据转储(data dump)　指定期将整个数据库复制到低速大容量的存储设备(磁盘阵列、光盘叠,磁带等)上保存起来。发生介质故障时将最近的一个转储副本调入,再利用其中的日志来完成故障恢复。

参考文献

1. 李昭原,等. 数据库技术新进展. 2 版. 北京:清华大学出版社,2007

2. 周龙骧. 数据库管理系统实现技术. 武汉:中国地质大学出版社,1990

(周龙骧　王翰虎　杨冬青)

shujuku guanjianci jiansuo

数据库关键词检索(keyword search in databases)　使用关键词对**数据库**内容进行检索的方法。

数据库关键词检索是数据库(database,DB)与信息检索(information retrieval,IR)交叉与融合的技术,主要研究如何在数据库中实现关键词查询、Top-k 检索、相关性排序、检索结果展现和相关性反馈等信息检索功能。数据库关键词检索使得用户不必懂得 SQL 查询,也不用了解数据库模式,通过简单的关键词查询就可以访问关系数据库,大大地提高了数据库的易用性。

在过去的几十年间,DB 和 IR 作为两个独立的计算机科学研究领域向前发展,都取得了巨大的成功。DB 的成功来自于 20 世纪 70 年代和 80 年代对商业数据处理的迫切需求,**关系数据库**成为存储和检索结构化数据的主要技术。IR 在过去的几十年间主要被应用于图书馆、资料库等的文档检索中。随着 90 年代互联网的兴起,基于信息检索技术的互

联网搜索引擎成为人们获取信息的主要方式,越来越多的数据库被置于互联网上供用户直接查询,使得数据库由面向应用变为面向用户,人们习惯于使用关键词查询来获取信息,也希望使用关键词查询来检索数据库。

近年来,数据库和信息检索都出现向对方领域渗透和融合的趋势,数据库领域的主要国际会议SIGMOD和VLDB和信息检索领域的主要国际会议SIGIR和WWW,都发表了大量DB和IR相结合的研究论文,也举行了各种形式的研讨会,涌现出各种各样的数据库关键词检索的实现方法和技术。

数据库关键词检索通常可以分为三个步骤,首先通过数据库全文索引检索到包含查询关键词的候选元组;然后搜索数据库模式图或数据图,把候选元组连接在一起,形成元组连接树作为检索结果;最后对检索结果排序并呈现给用户。数据库关键词检索主要研究内容包括数据模型、检索算法、结果排序和结果展现等方面。

数据库关键词检索系统可以分为在线查询和离线查询两大类。在线系统将关键词查询转换为SQL查询,通过数据库来实时生成查询结果。在线系统优点是不用很长的预处理时间,用户可以及时查询最新变化的数据,但其缺点是查询效率比较低。离线系统通过预先计算生成某种数据库的中间结果,当用户提交关键词查询时,根据中间结果生成查询结果。离线系统的优点是查询效率比较高,缺点一是需要比较长的预处理时间,二是预处理结果需要大量存储空间,三是用户难以及时查询数据库最新变化的数据。

参考文献

1. Wang S, Zhang K-L. Searching databases with keywords. J. Comput. Sci. & Technol., 2005, 20(1): 55-62

2. 王珊,文继荣. 数据库和信息检索技术的融合. 计算机学会通讯, 2006, 2(4): 4-12

(王珊　陈红　张俊)

shujuku guanli xitong

数据库管理系统(database management system,DBMS)　用于建立、使用和维护数据库的**软件系统**。它对数据库进行统一的管理和控制,以保证数据库的安全性和完整性。用户通过DBMS访问数据库中的数据,数据库管理员也通过DBMS进行数据库的维护。

按功能划分,数据库管理系统大致可分为6个部分:

(1) 数据定义语言(DDL)　用它书写的数据库模式被模块翻译为内部表示。数据库的逻辑结构、完整性约束和物理储存结构等描述信息被保存在DBMS内部的数据字典中。数据库的各种数据操作(如查找、修改、插入和删除等)和数据库的维护都是以数据库模式为依据的。

(2) 应用程序的编译　把包含了访问数据库语句的应用程序,编译成在DBMS支持下可运行的目标程序。

(3) 交互式查询　这部分模块提供易使用的交互式查询语言,如SQL。DBMS负责执行查询命令,并返回查询结果。

(4) 数据的组织与存取　该功能模块提供数据在外围储存设备上的物理组织与存取方法。这涉及三个方面:①提供与操作系统,特别是与文件系统的接口,包括数据文件的物理存储组织及内、外存数据的交换方式等;②提供数据库的存取路径及更新维护的功能;③提供与数据库描述语言和数据库操纵语言的接口,包括对数据字典的管理等。

(5) 事务运行管理　这部分功能模块提供事务运行管理及运行日志,事务运行的安全性监控和数据完整性检查和事务的并发控制及系统恢复等功能。

(6) 数据库的维护功能　为数据库管理员提供软件支持,包括数据安全控制、完整性保障、数据库备份、数据库重组以及性能监控等维护工具。

基于关系模型的数据库管理系统已日臻完善,并已作为商品化软件广泛应用于各行各业。它在客户/服务器结构的多用户环境中的应用,使数据库系统的应用进一步扩展。随着新型数据模型及数据管理实现技术的推进,可以预期DBMS软件的性能还将更新和完善,应用领域也将进一步地拓广。

Oracle是美国甲骨文(Oracle)公司开发的大型关系数据库管理系统,目前占有最大的市场份额,支持UNIX、Linux、Windows、Mac OS等多种操作系统。Oracle支持SQL语言,能在C、C++等主语言中嵌入SQL语句及过程化(PL/SQL)语句。应用程序可以通过使用开放数据库连接(ODBC)和Java数据库连接(JDBC)访问Oracle。

SQL Server是美国微软公司开发的关系数据库管理系统,支持Windows操作系统。SQL Server支持SQL语言,并进行部分扩充,成为Transact-SQL。

它采用图形用户界面,使系统管理更加简单。应用程序可以通过开放数据库连接(ODBC)、对象链接嵌入(OLEDB)和Java数据库连接(JDBC)访问SQL Server。

IBM DB2(简称DB2)是美国IBM公司研制的大型关系数据库管理系统,支持UNIX、Linux、OS/400和Windows等操作系统。DB2被公认为是最早使用SQL语言的数据库管理系统。它支持SQL和XQuery语言应用程序可以通过使用开放数据库连接(ODBC)、Java数据库连接(JDBC)或CORBA接口代理访问DB2。

Sybase是美国Sybase公司研制的大型关系数据库管理系统,支持UNIX、Linux和Windows等操作系统。Sybase提供了一套应用程序编程接口和库,可以与非Sybase数据源集成,适于创建多层应用,是一种开放的数据库管理系统。2010年,Sybase已被SAP收购。

MySQL是一个开放源码的小型关系数据库管理系统,开发者为瑞典MySQL AB公司,采用GNU通用公共许可证(GPL)。由于其体积小、速度快、成本低,特别是开放源码这一特点,许多中小型网站为了降低成本而选择MySQL作为网站数据库。

参考文献

1. Cardenas A F. Data base management systems. 2nd ed. Boston: Allyn Bacon Inc., 1985

2. Date C J. Introduction to data base systems. Volume Ⅰ,Ⅱ. 5th ed. Addison-Wesley Publishing Co., 1992

(李建中　许卓群　邹兆年)

shujuku lianjiexing biaozhun

数据库连接性标准(database connectivity standard)　不同的关系数据库管理系统(RDBMS)之间互访的接口标准。

20世纪80年代,结构化查询语言(SQL)成为**关系数据库**语言标准,但是在SQL标准中,没有提供一个公共的、与**数据库**无关的应用程序数据库访问接口。与此同时,RDBMS产品迅速发展。各个数据库厂家纷纷推出各自的SQL语言,在遵循标准的同时又都对标准SQL进行了不同程度的扩充或解释,使得不同的RDBMS提供的SQL互相不完全兼容。另外,不同厂商的RDBMS的客户系统与数据库服务器之间使用了不同的内部通信协议。因此,在某个RDBMS下开发的应用系统就难以访问在另一个RDBMS管理下的数据。现有的数据库连接性标准主要有开放数据库互连、Java数据库互连。

开放数据库互连(ODBC)是微软公司于1991年推出的为不同的RDBMS之间互访的接口,遵循1989年SAG制定的数据库调用级接口(DBCLI)草案。ODBC的发展得到了工业界和第三方软件商的支持,现在已经成为一种事实上的标准(并不是一种国际标准)。开放数据库互连(ODBC)的结构包括四个主要部分:应用程序、驱动程序管理器、数据库驱动程序和数据源,如图1所示。应用程序使用标准的数据库连接性标准,即应用程序接口(ODBC API)实现数据库的操作功能;驱动程序管理器为应用程序装载数据库驱动程序;当应用系统发出调用与数据源进行连接时,数据库驱动程序处理ODBC函数调用、向指定数据源发出SQL请求并将结果回送给应用系统。通过这四个部分使得应用程序和DBMS在逻辑上可以分离,应用程序与数据库无关,这样一个应用程序可以通过一组通用的代码访问不同的**数据库管理系统**。

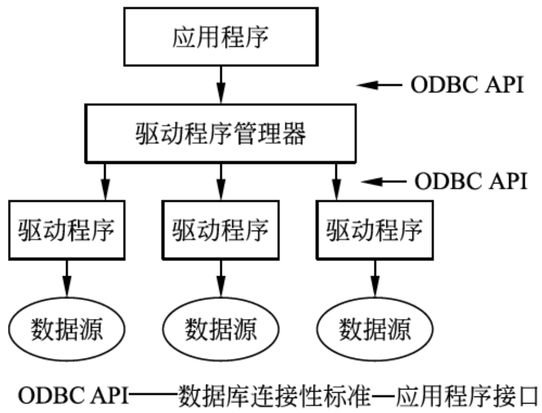


图1　开放数据库互连的标准结构

Java数据库互连(JDBC)是**Java语言**的数据库访问应用程序接口,是第一个标准的、支持Java的数据库应用程序接口。它使得Java程序与数据库连接更为容易。JDBC在功能上与ODBC相似,提供一个统一的数据库访问接口,因此可以用纯Java语言编写完整的数据库应用程序。实现对几乎任何一种数据库的访问。JDBC标准的推出使得在开发数据库应用时真正实现“一旦编写,到处运行”。

对象链接嵌入数据库(OLEDB)是美国微软公司设计开发的一种API接口,用于提供一种统一的方法来访问不同种类的数据源。OLEDB由数据提供者、数据使用者和服务组件三个部分组成。OLEDB作为ODBC的高级替代品,不仅支持关系数

据库上的应用,还支持面向对象数据库和电子表格等非关系数据库上的应用。

参考文献

1. Kyle G. Inside ODBC. Microsoft Press, 1995

2. Melton J. Understanding SQL and Java together: a guide to SGLJ, JDBC, and related technologies. Morgan Kaufmann, 2000

(李建中　顾宁　张绍华　邹兆年)

shujuku moshi

数据库模式(database schema)　对数据库中全部数据的逻辑结构和特征的描述。

1975 年,美国国家标准协会数据库管理系统研究小组设定了目前数据库领域公认的三级模式结构:外模式、模式和内模式。该结构能有效地组织、管理数据,提高数据库的逻辑独立性和物理独立性。

模式又称逻辑模式,是数据库中全部数据的整体逻辑结构描述。外模式又称子模式或用户模式,是从模式导出的一个子集,是与某一应用有关的数据的逻辑表示,是用户与数据库系统的接口。内模式又称存储模式,它是数据库中全体数据的内部表示,是数据库最低一级的逻辑描述,定义所有内部记录的类型、索引和文件的组织方式以及数据控制方面的细节。

数据库的三级模式是数据库在三个级别(层次)上的抽象,使用户能够逻辑地、抽象地处理数据而不必关心数据在计算机中的物理表示和存储。用户应用程序根据外模式进行数据操作,通过外模式——模式映射,定义和建立某个外模式与模式间的对应关系,将外模式与模式联系起来,当模式发生改变时,只要改变其映射,就可以使外模式保持不变,对应的应用程序也可保持不变;另一方面,通过模式——内模式映射,定义建立数据的逻辑结构与存储结构间的对应关系,当数据的存储结构发生变化时,只需改变模式——内模式映射,就能保持模式不变,因此应用程序也可以保持不变。从而提高了数据库的逻辑独立性和物理独立性。

参考文献

Imielinski T, Lipski W. A systematic approach to relational database theory. In: Proceedings of the 1982 ACM SIGMOD International Conference on Management of Data (SIGMOD '82). ACM, New York, NY, USA, 8-14. 1982

(汪卫　施伯乐)

shujuku sheji

数据库设计(database design)　根据用户的需求,设计数据库的结构和建立数据库的过程。在数据库设计过程中,产生了一系列的数据库模型,包括概念模型、逻辑模型和物理模型。

一般而言,数据库的设计过程可分为 5 个步骤:

(1)需求分析　调查和分析用户的业务活动和数据的使用情况,了解用户对数据库应用系统的要求,明确数据库应用系统所管理数据的种类、范围、数量以及它们在业务活动中交流的情况,确定用户对数据库应用系统的使用要求和各种约束条件等,形成用户需求规约。

(2)概念设计　对用户要求描述的现实世界(可能是一个工厂、一个商场或者一个学校等),通过信息的分类、聚集和概括等方式,建立抽象的概念数据模型。这个概念模型应反映现实世界各部门的信息结构、信息流动情况、信息间的互相制约关系以及各部门对信息储存、查询和加工等要求。所建立的模型应不涉及计算机上的具体实现细节,可以用一种抽象的形式表示出来。以扩充的实体—联系模型方法为例,第一步先明确现实世界各部门所含的各种实体及其属性、实体间的联系以及对信息的制约条件等,从而给出各部门内所用信息的局部描述(在数据库中称为用户的局部视图)。第二步再将前面得到的多个用户的局部视图消除冲突,集成为一个全局视图,即用户要描述的现实世界的概念数据模型。

(3)逻辑设计　主要工作是将现实世界的概念数据模型转换成数据库的一种逻辑模式,即适应于某种特定数据模型的逻辑数据模式。与此同时,可能还需为各种数据处理应用产生相应的逻辑子模式。关系型数据库已有一套较完整的规范化理论可用来部分地指导数据库逻辑设计。

(4)物理设计　根据特定数据库管理系统所支持的存储结构和存取方法等,对具体的应用任务选定合适的物理存储结构(包括文件类型、索引结构和数据的存放次序与位置等)、存取方法和存取路径等。

(5)验证设计　在上述设计的基础上,收集数据并具体建立一个数据库,运行一些典型的应用任务来验证数据库设计的正确性和合理性。

一般而言,一个大型数据库的设计过程往往需要经过多次循环反复。当设计过程中某步骤发现问题时,可能就需要返回到前面步骤重新修改。

参考文献

1. 王珊,萨师煊. 数据库系统概论. 4 版. 北京:高等教育出版社,2006

2. Silberschatz A, Korth Henry F, Sudarshan S. 数据库系统概念. 6 版. 杨冬青,李红燕,唐世渭,等译. 北京:机械工业出版社,2012

(何新贵　杨冬青　高军)

shujuku xitong

数据库系统(database system)　存储、管理、处理和维护数据的**软件系统**。它由**数据库**和有关软件组成。这些软件包括**数据库管理系统**(DBMS)、宿主语言、开发工具和应用程序。DBMS 用于建立、使用和维护数据库。*宿主语言*是可以嵌入数据库语言的程序设计语言。数据库是长期储存在计算机中有组织的、大量的和可共享的数据集合。

数据库系统的学科含义是指研究、开发、建立、维护和应用数据库系统所涉及的理论、方法和技术。数据库系统是软件领域的一个重要分支,涉及计算机应用、软件和理论三个方面。

数据库系统的发展主要以数据模型和 DBMS 的发展为标志。它诞生于 20 世纪 60 年代中期。第一代数据库系统是指层次和网状数据库系统。其代表是 1969 年美国 IBM 公司研制的层次数据库系统 IMS 和美国数据库系统语言协会 CODASYL 的数据库任务组(DataBase Task Group,DBTG)于 20 世纪 60 年代末至 70 年代初提出的 DBTG 报告。DBTG 报告基于网状模型提出了数据库系统的许多概念、方法和技术。第二代数据库系统是指关系数据库系统。1970 年 IBM 公司 San Jose 研究所的 E. F. Codd 发表了题为"大型共享数据库的关系模型"的论文,开创了关系数据库系统方法和理论的研究。20 世纪 90 年代随着面向对象、人工智能和网络等技术的发展,产生了面向对象数据库系统和演绎数据库系统。近年来随着数据库应用领域的拓展,在 Web 数据管理和生物数据管理等应用的推动下,半结构化和非结构化数据库成为主要的发展方向。

数据库中的数据有逻辑和物理两方面特性。逻辑方面由全局模式(模式)和子模式(外模式)描述;物理方面由存储模式(内模式)描述。这三个模式是对数据库的三级抽象,即数据库的三级结构。三级结构的采用使数据库具有较高的物理独立性和一定的逻辑独立性。数据库中的数据包括:数据本身、数据描述、数据约束和数据的存取路径。数据库独立于具体的应用,为多个应用共享,从而大大减小了数据的冗余度和不一致性。

数据库由 DBMS 统一管理,数据库的建立、使用和维护均要通过 DBMS 进行。建立数据库时,DBMS 将用数据定义语言(DDL)写的数据库模式翻译成内部表示,数据库的逻辑结构、完整性约束和物理结构等描述信息保存在数据字典中;使用数据库是通过 DBMS 提供的数据操纵语言(DML)或查询语言进行(程序方式或交互方式);DBMS 提供的维护数据库的功能包括数据的安全性控制、完整性保障、并发控制、数据库恢复和数据库重组。同计算机系统的最新成果相结合是当前数据库管理系统技术的主要特点,当今的 DBMS 的设计考虑了 Flash、多核、GPU 和云计算平台等技术特点。

参考文献

1. Ullman J D. Principle of database and knowledge-base system. Vol. Ⅰ, Ⅱ. Computer Science Press, 1989

2. 施伯乐,等. 数据库系统导论. 北京:高等教育出版社,1994

(施伯乐　朱扬勇)

shujuku xitong sanji jiegou

数据库系统三级结构(three-level architecture of database system)　数据库系统的一种结构框架,它由外部层(单个用户的视图)、概念层(全体用户的公共视图)和内部层(存储视图)组成(见图 1)。该结构由美国 ANSI/X3/SPARC 的 DBMS 研究组于 1975 年提出,现在大部分数据库产品都支持该结构框架。数据库系统三级结构的目的是将用户视图、数据模式以及底层的存储方式分隔开来。

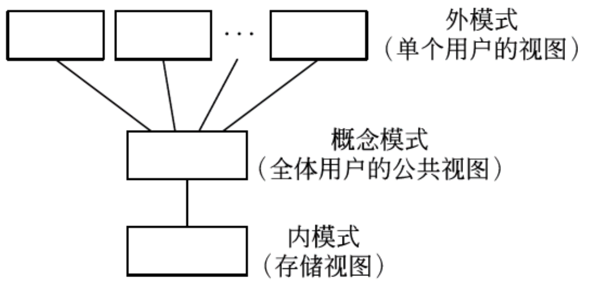


图 1　数据库系统三级结构示意图

外部层是最接近用户的层次。它是数据库的"外部视图",是各个用户所看到的数据库。它所表示的是面向单个用户的数据库的局部逻辑。

内部层是最接近物理存储的层次。它是数据库的“内部视图”或“存储视图”。它与数据库的实际存储密切相关,可以理解为机器“看到”的数据库。

概念层是介于上述两者之间的层次。它是数据库的“概念视图”,是数据库中所有信息的抽象表示。它既抽象于物理存储的数据,又区别于各个用户所看到的局部数据库。概念视图可以理解为数据库管理员所看到的数据库。

数据库系统结构的外部层、概念层和内部层分别对应于数据库模式的外模式、模式和内模式。

数据库系统结构分级对于提高数据独立性具有重要意义。在三级结构间存在着两级映射。概念层/内部层映射定义了概念视图与数据库存储结构之间的对应关系。如果数据库的存储结构发生变化,可以相应地改变概念层/内部层映射,从而使概念视图保持不变,即将数据库存储的变化隔离在概念层之下,这称作**数据的物理独立性**。外部层/概念层映射定义了单个用户的外部视图与全局的概念视图之间的对应关系。如果概念视图发生变化,可以改变外部层/概念层映射,从而使用户看到的外部视图保持不变,避免对应用程序的影响,这称作**数据的逻辑独立性**。

参考文献

1. Date C J. 数据库系统导论. 8 版. 孟小峰, 王珊, 姜芳艽, 译. 北京: 机械工业出版社, 2007

2. 王珊, 萨师煊. 数据库系统概论. 4 版. 北京: 高等教育出版社, 2006

（*唐世渭　杨冬青*）

shujuku xingneng ceping

数据库性能测评（database performance evaluation）　在特定软、硬件环境下(包括特定工作负载条件下)运行数据库管理系统(database management system),获得被测系统(system under test,简称 SUT)运行效率相关的测试数据,以对数据库产品的性能表现及其运行支撑环境的成本进行综合分析和评价的过程与方法。

在评价一个数据库产品性能前,首先要建立合适的性能测评指标体系,并设计测量指标及其综合指标的计算方法。依照 GB/T 16260《软件产品-质量模型》,数据库产品的性能测评包括时间特性、资源特性和依从性三个方面。时间特性指标是指针对 DBMS 的一些长响应时间操作(如数据导入/导出、数据库备份/恢复)开展测试,获得响应时间、等待时间、处理时间及系统吞吐能力等性能值。资源特性指标是针对 DBMS 在不同应用场景下所能承受的系统资源负荷开展测试,获得数据库系统的 I/O 设备、内存资源及网络占用等系统资源利用测量值,从而发现系统资源的性能瓶颈。依从性指标是参照特定的性能基准(如 TPC-C 性能基准)的数据模型、工作负载及其测试规范而获得数据库系统的性能指标最佳值,从而为用户更好地选择和应用 DBMS 产品提供依据。

数据库性能测评就是针对具体的性能测评指标体系设计测试数据模型、测试数据特征及测试工作负载,规定测试数据生成方法、测试负载控制机制等过程与方法。面对不同的评价目标,数据库性能测评分为三种情况:①面向某个具体应用的数据库性能测评　数据库厂商与应用程序开发商配合在某个具体应用环境中运行一个数据库系统,以获得包括系统是否稳定可靠、使用是否方便、服务是否周到、配置是否完备、价格是否合理等实际应用的综合性能测量数据;②面向某类行业应用的数据库性能测评　用户定义一组含有实际行业应用特征的应用基准程序,如 SAP 公司的面向 ERP 的性能基准测试程序、Oracle 公司的面向商务智能应用的性能基准程序等,通过行业典型应用基准组合的测试来反映应用系统的实际使用情况,以帮助用户进行数据库产品及其运行支撑硬件的选型。因此面向行业应用的数据库性能测评适合对垂直行业应用的 DBMS 及其服务器进行评价。③基于基准的数据库性能测评　基准测试是指依据某个公认性能标准,设计测试程序,搭建测试环境,对被测数据库系统依照测试规范进行性能测试。基准测试一般是检测一个部件或系统可以达到的最佳性能。因为基准测试规定了相对严格的测试标准,所以利用测试结果可以对不同的 DBMS 进行比较。数据库厂商或硬件厂商就可利用基准测试结果发现自身系统的优势与劣势,针对自身产品优势进行市场宣传和定位;用户可利用基准测试结果指导数据库产品及其运行支撑硬件产品的选型。

人们普遍关注的是基于基准的数据库性能测评,这也是信息系统领域应用最广泛、实施最成功的数据库管理系统性能测评方法。20 世纪 80 年代中后期,数据库企业对基于基准的数据库性能测评标准展开了激烈的论战,在 Omri Serlin 先生的协调下 8 家数据库相关企业联合发起了事务处理性能委员会(Transaction Processing Performance Council),致力于 TPC 系列性能基准规范的制定与推广。另外一

类性能基准是由标准性能评估机构(Standard Performance Evaluation Corporation)发布的SPEC系列性能基准。SPEC基准关注的是服务器硬件CPU和内存的性能以及Web应用中Java企业应用服务器的性能,而TPC基准注重于数据库管理系统的事务处理和分析挖掘能力,测试针对的是数据库管理系统的最佳性能。目前TPC系列性能基准已成为评价数据库性能的事实标准,是最常用、最普及的度量数据库产品的整体性能及其性价比的工业标准。

参考文献

Gray Jed. The benchmark handbook for database and transaction processing systems. 2nd ed. San Mateo, CA: Morgan Kaufmann, 1993

(叶晓俊　王建民)

shujuku yiqi

数据库移栖(database migration)　将数据库转移到新环境的机制和过程。数据库管理系统软件升级、数据库在网络中的动态定位以及某一数据库管理系统的应用程序使用第三方数据库等几种情况需要将数据库转移到新的环境中。新环境中的数据库管理系统提供移栖机制将要移栖的数据库(源数据库)转移到新的环境(目标数据库)。主要过程如下:首先分析源数据库模式与目标数据库模式的差异,然后用移栖机制提供的转换功能转换源数据库模式,最后把源数据库中的数据按转换后的模式复制到目标数据库。

参考文献

Hara T, Harumoto K, Tsukamoto M, et al. Database migration: a new architecture for transaction processing in broadband networks. IEEE Trans. on Knowledge and Data Engineering archive, 1998, 10(5)

(周龙骧　王翰虎)

shujuku yingyong jishu

数据库应用技术(database application technology)　在数据库应用系统的开发过程中逐渐形成的数据管理和利用方面的技术。

主要包括:

基于新型计算平台的数据处理技术

随着分布式计算技术和云计算技术的发展,以Google/Hadoop为代表的新型分布式计算平台技术日益成熟,催生了基于新型计算模型的海量数据的分析技术。它以GFS、HDFS等分布式文件管理系统为基础平台对海量数据进行管理和并行分析处理,实现利用大量廉价计算资源进行海量数据处理的目标。

新型的数据类型管理技术

当前信息系统管理的数据类型多种多样,对数据处理的方式也各不相同,从而出现了各种针对不同类型数据的数据库应用系统。例如万维网数据管理技术是针对海量万维网数据管理与检索的需要产生的。社会网络系统则是针对当前大量的社会网络数据的组织与管理需求产生的。针对人们在物理、化学、生物、天文、地理等学科的科学研究过程中生成的大量数据的管理诞生了科学数据库技术。

新的数据应用和管理模式

信息集成　对分布、异构的系统进行信息和服务的集成是新型数据库应用系统中的主要内容之一。信息集成的模式主要有两种:

(1) 数据集成的方式,其代表技术是数据仓库技术。

(2) 模式集成的方式,其代表技术是数字图书馆技术。

在这两种模式中,均需要解决数据质量评估和修正以及信息抽取等关键问题。

数据深层分析　随着信息系统技术的发展,各类系统中积累了大量的数据,对数据的深层次分析成为数据库应用技术的新的发展方向。目前对数据进行深层次分析主要有两种方式:

(1) 联机分析处理　它通过对海量的数据进行有效地组织,以支撑管理决策人员对数据从各个不同的角度或角度的组合进行深层次的分析。目前联机分析处理主要包括基于关系数据库和基于多维数组两条技术途径。

(2) 数据挖掘　数据挖掘是从大量的、不完全的、有噪声的、模糊的、随机的实际应用数据中,提取隐含在其中的、人们事先不知道的,但又是潜在有用的信息和知识的过程。

新的数据管理和操作　随着数据利用的方式越来越多,出现了很多对数据库的新操作。例如随着搜索技术的广泛使用,针对数据库系统的关键词搜索成为研究的热点;随着人们对隐私的重视,数据隐私保护成为数据库应用中新的数据保护模式;随着数据库系统的不断升级换代,数据库的移栖成为数据库领域的新问题。

参考文献

1. Han J W, Kamber M J, Pei J. Data mining: concepts and techniques. Morgan Kaufmann Publisher, 2011

2. Ramakrishnan R, Gehrke J, Database management systems. 3rd ed. McGraw Hill Higher Education, 2002

（汪卫　施伯乐）

shujuku yuyan

数据库语言（database language）　满足用户对结构化数据或者半结构化数据的定义和查询等需求的计算机语言。

从数据类型的角度，数据库语言的发展可以从结构化数据查询语言（structured query language，简称 SQL）和半结构化数据查询语言两个方面阐述。

结构化数据查询语言　SQL 已经成为关系数据库的国际标准语言，它的原名叫做 SEQUEL，是在 Boyce 等人于 1975 年提出的 SQUARE（specifying queries as relational expressions）语言的基础上发展起来的。后来 Boyce 等人不断对 SQUARE 语言的语法进行修改，形成了 SEQUEL 语言。SEQUEL 语言最早是 IBM 的 San Jose 研究室（Almaden 研究中心，请参阅网站 http://almaden.ibm.com）在 20 世纪 70 年代初作为 System R 项目的一部分而实施的。1976 年，Chamberlin D D 等人进一步对 SEQUEL 语言进行了改进，最后形成了 SQL 语言。由于 SQL 功能丰富、语言简捷，倍受用户及计算机工业界欢迎，被众多计算机公司和软件公司所采用。经各公司的不断修改、扩充和完善，SQL 语言最终发展成为关系数据库的标准语言。1986 年 10 月美国国家标准协会 ANSI 的数据库委员会 X3H2 批准了 SQL 作为关系数据库语言的美国标准。同年公布了 SQL 标准文本（简称 SQL-86）。1987 年国际标准化组织 ISO 也通过了这一标准。此后 ANSI 不断修改和完善 SQL 标准，并于 1989 年公布了 SQL-89，1992 年又公布了 SQL-92 标准。目前 ANSI 公布的新标准是 SQL-99 或 SQL-1999，亦称 SQL3。

SQL 语言除了具有查询数据库的功能以外，还具有定义数据结构、修改数据和说明安全性约束条件等特性。它主要由以下几个部分组成：

（1）SQL DDL　定义关系模式、删除关系、建立和删除索引以及修改关系模式；

（2）SQL DML　查询、插入、删除和修改数据；

（3）嵌入式 DML　将 SQL 语句嵌入到 PASCAL、C 等宿主语言中；

（4）视图定义　主要用于创建视图；

（5）权限管理　对关系和视图的访问进行授权；

（6）完整性　定义数据必须满足的完整性约束条件；

（7）事务控制　定义事务的开始、提交和结束等。

SQL 是一种介于关系代数和关系演算之间的语言，是一种非过程化的集合操作语言。它具有关系代数和关系演算的双重特点：①综合统一；②高度非过程化；③面向集合的操作方式；④以一种语法结构提供两种使用方式：SQL 既是自含式语言，又是嵌入式语言；⑤语言简捷，易学易用。

半结构化数据查询语言　XML 的全称是 extensible markup language（可扩展标记语言），它是一种专门为 Internet 设计的标记语言。1998 年 W3C 发起了查询语言专题讨论会，重点研讨 XML 查询语言的重要特性和需求，这就是著名的 XQuery 规范。XQuery 是 XML query language 的简称，目前的版本是 1.0，它是截止到 2003 年 11 月 12 日的 W3C 工作草稿。W3C 的 XQuery 是个非常复杂的规范，目前包含了 12 个不同的工作草案（而且可能还会增加）。该规范当前正在迈向推荐标准状态。XQuery 起源于 quilt，前身是 XQL。XQL 是目前 XML 查询语言世界中最接近实际标准语言的一种语言，已经有十几种实现在使用。对于 XML 用户来说，最熟悉的 XQuery 关键组件是 XPath，它本身就是 W3C 的一个规范。XPath 是一些有关如何在 XML 文档中进行定位，即如何很快找出 XML 文档中具有某种特征标签的一种语言。例如，如何在一个 XML 文档中找出第 5 次出现的 <Book> 标签所在的节点。

参考文献

1. Ullman J D. Principles of database systems. 2nd ed. London: Pitman Publishing Ltd., 1983

2. 王珊，萨师煊. 数据库系统概论. 4 版. 北京：高等教育出版社，2006

3. 冯建华，周立柱，郝晓龙. 数据库系统设计与原理. 2 版. 北京：清华大学出版社，2007

（冯建华）

shuju leixing

数据类型（data type）　数据对象的取值集合以及对之可施行的运算集合。数据类型规定了具有该

类型的变量或表达式的取值范围,也规定了与之相联系的运算的集合。

几乎每种**程序设计语言**都具有**简单类型**、**构造类型**和**指针类型**。某些程序设计语言还具有递归类型,它是借助其自身定义的类型,即在其类型定义的右方包含该类型本身的类型。递归类型的值集由借助相应类型定义方式(右方该类型的初值为空集)依次递推而得到的各个值组成。

在静态类型的语言中,所有变量和参数的类型都由程序员确定,这样每个表达式的类型都可以在编译时刻确定,所有操作都可以在编译时刻进行类型检查,大部分**高级语言**是静态类型语言。在动态类型的语言中,只有值的类型是确定的,变量和参数没有指定的类型,它们在运行中的不同时刻可取不同的值,这表明,在运行时执行一个运算之前必须对运算分量进行类型检查。LISP 和 Smalltalk 是动态类型语言的例子。

在进行类型检查时,必须判别两个类型 T 与 T′ 是否等价,有两种判别类型等价的方法,即按名等价与按结构等价,它们的定义是:

按名等价:T≡T′　当且仅当 T 和 T′在同一处中定义

按结构等价:T≡T′　当且仅当 T 和 T′具有相同的值集

考虑下列说明:

```
TYPE   T1 = Record     T11: Integer;
                       T12: Character
          End;
       T2 = Record     T21: Integer;
                       T22: Character
          End;
var       f1: T1;
          f2: T2;
procedure P(var f:T1);
```

过程调用 P(f2)在按名等价方式下不能通过类型检查,因为 f2 的类型 T2 与 P 的参数类型 T1 在不同的说明中定义,但在按结构等价方式下可以通过类型检查。

由此导致类型定义的两种认识,在按结构等价的语言中,类型定义仅用于建立标识符与已有类型间的**绑定**,而在按名等价的语言中,类型定义实质上创建了一个新的类型。

参考文献

Watt D A. Programming language concepts and paradigms. Prentice Hall,1990　(金凌紫　陈涵生)

shuju lianluceng

数据链路层(data link layer)　OSI 开放系统互连基准(参考)模型中的第二层,介于物理层和网络层之间。在 TCP/IP 参考模型中属于链路层的主要部分。

数据链路层在物理层提供的数据帧传输服务的基础上向网络层提供数据包传输服务,其主要任务是将源节点网络层的数据包可靠地传输到相邻目标节点的网络层。数据链路层使用的协议称为**数据链路协议**(data link layer protocol,或 data link protocol)。

数据链路层的主要功能有:

(1) 成帧　网络层的数据包被分割成独立传输的数据块(数据帧,frame),并加上相应的控制信息(如地址、顺序号、检错码等)。

(2) 流量控制　调节发送速率与接收方相匹配,防止发送节点发送过多数据帧,导致接收节点数据溢出。

(3) 差错控制　检测和纠正数据帧在传输过程中可能发生的差错。例如,通过检测接收节点数据帧中的检错码判定传输错误后,根据使用的服务类型,进行相应的处理,如果是可靠服务,接收帧被重传;而对于不可靠服务,则接收帧被丢失并且依赖上层协议处理该问题。

(4)在多址共享介质上寻址　在共享传输介质的环境中,识别发送节点和接收节点的地址。

(5)链路管理　两个网络实体(源节点和相邻目标节点)之间提供数据链路的建立、维持和释放的管理。

数据链路层使用的协议称为数据链路控制协议,也称链路通信规程,链路控制协议可分为异步协议和同步协议两大类。常用的数据链路协议(data link protocol)有:

(1) HDLC 高级数据链路协议 (high-level data link protocol)　重要的数据链路控制协议,是其他协议的基础和来源。

(2) LAP 链路访问规程　LAPB 提供了在 X. 25 数据分组交换网络上点对点连接的协议。LAPD 提供了对于综合业务数字网 ISDN 连接的 D 信道的数据链路控制。LAPF 为帧中继网络提供了数据链路。

(3) ATM 异步传输模式(asynchronous transfer mode)　具有基于信元的简单数据链路层协议。

(4) PPP 点对点协议(point-to-point protocol) 主要面向电话拨号链路,传输 IP 或其他协议的数据分组。PPP 帧包含了识别相应协议(IP, IPX 等)类型的字段。

(5) LLC(逻辑链路控制) 是 IEEE 802. x 网络标准系列中的一部分,该系列包含了主要的局域网标准:以太网、快速以太网、FDDI 和令牌环等。IEEE 802. x 网络标准将数据链路层分为介质访问控制(MAC)和逻辑链路控制(LLC)两个子层(sub-layer)。LLC 子层与 HDLC 协议相似,提供数据链路控制功能。较低的 MAC 层定义了不同传输介质的接入方法,例如以太网(Ethernet)的载波监听多路访问/冲突检测(CSMA/CD)等。

随着计算机网络技术的发展,新的数据链路层技术不断出现,例如 MPLS 多协议标签交换(multi-protocol label switching)技术、SDN 软件定义网络(software-defined networking)中的 OpenFlow 协议等。

参考文献

1. Forouzan B A. 数据通信与网络. 2 版. 吴时霖, 周正康, 吴永辉, 等译. 北京: 机械工业出版社, 2002

2. Gilbert Held. 数据通信. 6 版. 戴志涛, 卞佳丽, 郑岩, 译. 北京: 人民邮电出版社, 2000

(许勇　张凌)

shujuliu jisuanji

数据流计算机(dataflow computer)　以数据驱动方式进行操作的计算机。它是一种在指令操作的驱动原理上与传统的冯·诺依曼计算机根本不同的**非传统计算机**。

在传统的冯·诺依曼计算机中,指令执行的次序是按照指令在程序中规定的顺序进行的,因此被称为控制驱动的计算机。控制驱动计算机中的指令计数器从本质上规定了指令的串行执行,另外还规定多条指令可共用一个存储单元中的操作数,因而会产生副作用,引起指令间的数据相关,也影响指令的并行执行。数据流计算机取消了指令计数器和存储器的数据共享。在数据流计算机程序中的任一条指令,只要所需的操作数已全部齐备,就可以立即执行,一条指令的运算结果又流向下一条指令作为其操作数来启动该指令的执行,这就是*数据驱动*。这样做可以使所有有条件执行的指令都能并行执行。如果在硬件上有足够多的执行部件、路由选择网络等资源,就能最大限度地实现指令级的并行执行,使并行处理能力高于冯·诺依曼计算机。

数据流计算机的程序适合于用有向图来表示,称为数据流图。图 1 是计算 $a=(b+c)+(d\times e)$ 的数据流图。图中的每一个节点表示一个操作,操作可以是一条指令,也可以是一组指令或一个功能块。节点之间通过有向弧连结。在数据流图中流动的数据,包含具体数值和该数据的流向信息,这种形式的数据称为数据标记。数据流图中还有带有布尔值的控制标记。当节点的所有输入弧上都有数据标记或控制标记时,便可执行操作,而不需指令计数器。执行后输入弧上的标记消失,运算结果出现在输出弧上,流向后继节点去起动下一个操作,这样便没有共享数据。

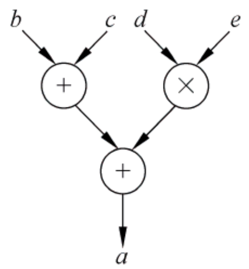


图 1　数据流图举例

显然,按上述驱动原理操作,必然要执行很多不必执行的指令,例如条件转移时,两个分支的指令都要执行,这样便加重了处理部件、路由选择网络等硬件的负担,从而降低系统的效率。在后来的一些数据流计算方案中对此作了改进,使操作中只有选定数量的操作数或函数的独立变量才能用来启动操作的执行,这样可以避免不必要的计算工作。

数据流计算机有静态和动态之分。在静态数据流计算机中,每条输入弧上只允许有一个数据标记存在,不允许对一个节点作并行的多次调用,禁止对可重入代码的并行处理。它的数据流图在编译时就已完全确定,所以称为静态。这种方式实现起来简单,但限制了并行性的开拓。在动态数据流计算机中,对每个数据标记添加一个特定的标志,使每条输入弧上的数据标记可以多于一个,这样可以并行调用可重入代码,有利于并行性的开拓。它的数据流图在执行过程中是有变化的,所以称为动态。还有一种动态数据流计算机,它不加标志,由代码复制来实现可重入代码的并行调用。这种系统可以同时存在一段可重入代码的多个副本,但要增加系统的开销。另外,还曾提出过一些不同于静态和动态数据流计算机的特殊系统结构数据流计算机。还有一些

是混合驱动模式的，有的采用数据流与控制流的混合，有的采用数据流与需求流的混合。

20世纪60年代末美国麻省理工学院的Dennis及其研究组开始研究叫作VIM的VAL语言解释机，在数据流图、数据流计算语言和数据流计算机系统结构等方面做了开创性的工作。从70年代后期到80年代，数据流计算机的研究在世界上形成了热潮。研制的数据流计算机最初主要用于数值计算，后来也研制了专门用于符号处理的非数值计算的机器。这一时期有影响的项目，在静态数据流机方面有美国的Taxes分布式数据处理机、法国的LAU系统、日本的NEDIPS和IPP系统等，在动态数据流机方面有美国的MIT带标记的数据流机和DDM-1机、英国曼彻斯特数据流机、日本的SIGMA-1系统等，在非数值计算的数据流机方面有日本的EM-3机、DFM机（原始样机为EDDY）、PIM-D机和TOPSAR机等。在澳大利亚、印度等国也有研究项目。研制的机器或原始样机绝大多数是作为传统的控制驱动计算机的附加处理机，由控制驱动计算机作为主机来处理译码和输入输出任务。

在数据流计算机的研究项目中，只有极少数做出能运行的机器，大多数只做出原始样机，还有的仅仅是在传统计算机上做些模拟和分析工作。第一个能运行的机器当属DDM-1机，其他能运行的还有EDDY、曼彻斯特数据流机、LAU系统等。对于数据驱动的优点，有的还只是设想，仅经过性能分析和模拟试验，做成的一些原始样机一般也只能提供在有限条件下的实际运行分析数据。

在进入20世纪90年代后，传统计算机从性能、并行处理技术和性能价格比等方面都有长足进展，在市场上占有很大优势，数据流计算机始终无法形成产品，进入市场。现实情况促使研究人员改变纯数据流的设想，与计算机生产厂商合作研究开发混合结构的设计方案，使系统结构尽量接近传统计算机，并采用传统计算机现成的硬件和软件子系统，包括现成的微处理机。但数据流计算机的研究工作在90年代上半期仍然不得不陆续停下来。（孙强南）

shujuliu tu

数据流图（data flow diagram）　一种用于表示数据在系统运行过程中的流向与处理（加工）的图形工具。包含五个基本成分：数据流、处理、数据存储、数据源和数据宿。“数据流”规约了系统中的数据及其流动。数据流名通常是名词或名词短语，例如“学生成绩”；“处理”表示数据的变换，用于规约系统的功能。处理名通常采用动宾结构，例如“统计学生平均成绩”；“数据存储”是数据静态结构的表示，“数据源”和“数据宿”分别是数据的产生地和归宿，它们定义了系统的边界。

数据流图作为一种系统模型的表示工具，在使用中，支持数据抽象和过程抽象；并提供分层结构，支持自顶向下逐步精化地建造系统模型。例如在结构化方法中，首先建立系统的顶层数据流图，即系统环境图，而后，通过对上一层数据流图的不断精化，得到一个可用的系统模型的表示，其中每一处理表示一个明确的功能。（王立福）

shujuliu wajue

数据流挖掘（data stream mining）　对数据流的挖掘（参见**数据挖掘**）。数据流是指在实时监控系统、互联网等众多应用中产生的高速、海量的数据序列。由于数据流的规模是潜在无限的，无法在内存中存储整个数据流以多次执行扫描操作，因此数据流挖掘通常只运用单次扫描数据的方式来获得挖掘结果。在传统的数据挖掘方法中，全部数据都存储在计算机系统之中，且能被多次扫描，这些方法无法用于挖掘数据流。此外，数据流挖掘与传统数据挖掘之间还存在其他显著的差异：数据流挖掘要求实时处理数据，传统数据挖掘则通常离线处理数据；数据流挖掘返回近似结果，传统数据挖掘则要求获得准确的结果。

由于无法在内存中保存所有数据，数据流挖掘通常需要在内存中维护一个紧凑的数据结构来描述数据流，并据此计算挖掘结果。这类数据结构被统称为大纲（synopses），其空间开销远低于数据流的规模。大纲仅维护数据流的概要信息，无法用于计算精确结果，只能计算近似结果。通常情况下，大纲所占用的空间越大，近似结果的质量就越高。令N表示数据流的规模，数据流挖掘一般采用对数阶空间复杂度$O(\log^k N)$来描述大纲，而非$O(N)$。为了满足实时性要求，处理单个数据的平均时间复杂度也是$O(\log^k N)$。常用的大纲数据结构和方法包括：采样、直方图、平衡二叉树、小波、随机算法等。随机算法一般能够确保近似结果的高置信度。由于数据流中的各项数据按时间顺序到达，因而数据流挖掘还需考虑与时间相关的模型，包括仅考虑最近W个元组的滑动窗口模型和元组重要性随时间减弱的衰减窗口模型等。

数据流挖掘并不局限于处理典型的数据流，它也已经成为传统数据挖掘技术的重要补充。随着数据规模的不断扩大，传统的需要多次扫描数据集合的挖掘技术的开销变得越来越昂贵。相比之下，数据流挖掘技术仅仅单次扫描数据集合，这样不但能显著降低处理开销，而且其高质量的近似结果在多数情况下也可满足应用需求。数据流挖掘因为其处理实时数据的能力，在传统的实时数据库的应用中也引起关注。

参考文献

1. Han J，Kamber M，Pei J. Data mining：Concepts and techniques. 3rd ed. Morgan Kaufmann，2011

2. Aggarwal C. Data streams：Models and algorithms. Berlin：Springer，2007　（金澈清　周傲英）

shujuliu yuyan

数据流语言（data flow language）　一类专门用于数据流计算机的**函数式程序设计语言**。其计算模型的基本原理主要有两条：①数据驱动。当且仅当指令所需的数据可用时（值全部到达，且不存在数据相关性），该指令即执行操作；②任何指令操作均为纯函数操作。

数据流语言遵循单赋值规则，一个变量只能在赋值语句的左部出现一次。数据流语言没有函数副作用，因此，发现并行性比较容易。实际上，数据流语言和逻辑式程序设计语言是可在编译时通过数据相关性分析和全局流分析发现隐并行性的两类重要语言。

用于数据流计算机的语言可分为两类。一类是数据流机的机器语言，常称为数据流图（DFG）；另一类是数据流机的高级语言，常称为数据流语言。

DFG 是一种有向图。任何一个计算任务都可用 DFG 描述。例如，计算 $(a/b)+(a*b)$ 的 DFG，如图 1 所示。

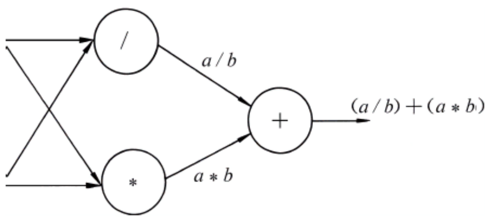


图 1　计算 $(a/b)+(a*b)$

DFG 直观，易于理解，刻画了数据流的一般原理，但不便于用户使用。

数据流语言是完全基于 DFG 的高级语言，追求以自然方式表达最大的并行性，其主要设计目标是：①提供隐式并行，以表达问题的内在并行性；②很易转换为高度并行的 DFG，在数据流机上执行；③清晰简明，模块化，有助于生成正确的程序。

数据流语言主要特点是：①是面向值的系统，抛弃了“变量”、“存储”等概念，消除了副作用。语言的活动部分（表达式，函数）具有不相干性，一旦所有输入值均为已知，则函数和表达式的执行不可能影响任何其他拟执行操作的结果；②极强的并行表达能力；③良好的“作用局部化”性质，使指令之间没有不必要的数据相关性，并可大大简化转换 DFG 的工作。

数据流语言的主要缺点是缺乏递归和没有通用的输入输出机制。

数据流语言的典型代表是 MIT 的 J. B. Dennis 和 W. B. Ackerman 设计的 VAL，加州大学 Arvind 和 K. P. Gostelow 设计的 ID 以及法国 D. Comte 等设计的 LAU。

参考文献

Ackerman W B. Data flow languages. IEEE Computer，1982，15(2)　（郭浩志）

shuju moxing

数据模型（data model）　数据库领域中定义数据及其操作的一种抽象表示。

数据模型由三部分组成：① 实体及实体间联系的数据结构描述；② 对表示实体和联系的数据的操作；③ 完整性约束。

数据结构描述数据库的组成对象以及对象之间的联系。它是所描述的对象类型的集合，是对系统静态特性的描述，是刻画一个数据模型性质最重要的方面。在数据库系统中，人们通常按照其数据结构的类型来命名数据模型。

数据操作是指对数据库中各种对象（型）的实例（值）允许执行的操作的集合，包括操作及有关的操作规则。数据库主要有查询和更新（包括插入、删除、修改）两大类操作。数据模型必须定义这些操作的确切含义、操作符号、操作规则（如优先级）以及实现操作的语言。数据操作是对系统动态特性的描述。

完整性约束是一组完整性规则，它表达了给定的数据模型中数据及其联系所具有的制约和依存规则，用以限定符合数据模型的数据库状态以及状态

的变化，以保证数据的正确、有效和相容。

根据模型应用的不同目的，数据模型可分两大类。它们分别属于两个不同的层次。第一类模型是面向应用的，称为**概念模型**，也称信息模型。它是按用户的观点来对数据和信息建模，主要用于**数据库设计**。这类数据模型描述用户和设计者都能理解的信息结构，强调其表达能力和易理解性。如 ER 模型（参见实体联系模型）。第二类模型是面向**数据库管理系统**的，它是按计算机系统的观点对数据建模，主要用于数据库的实现。第二类中的逻辑模型用以刻画实体在数据库中的逻辑表示。例如，层次模型（hierarchical model）、网状模型（network model）、关系模型（relational model）、面向对象模型（object oriented model）、对象关系模型（object relational model）、XML 模型以及非结构化的 key-value 模型等。第二类中的物理模型刻画数据在数据库中的存储。它描述数据在系统内部的表示方式和存取方法，在磁盘、光盘上的存储方式和存取方法，是面向计算机系统的。

数据模型是数据库系统的核心和基础。各种 DBMS 软件都是基于某种数据模型的。

数据模型的研究与实现无论在过去、现在和未来，一直是数据管理以及相关领域一个重要而困难的科学问题。数据的多样性、复杂性、应用的差异性以及计算机系统的层次性、开放性和复杂性，决定了数据模型的多样性、复杂性和差异性。如何对数据建模，如何根据数据的属性、数据的语义以及应用的需求，建立科学的合适的数据模型是问题的难点所在。

参考文献

1. Ullman J D. Principles of database systems. 2nd ed. London: Pitman Publishing Ltd., 1983

2. 王珊，萨师煊. 数据库系统概论. 4 版. 北京：高等教育出版社，2006

3. “10000 个科学难题”信息科学编委会. 10000 个科学难题. 信息科学卷. 北京：科学出版社，2011

（王珊　陈红）

shuju tonglu

数据通路（data path）　计算机中的数据、地址、指令等信息在机器内部的传送路径。用于表示计算机数据单元（包括寄存器和存储器）与操作单元（包括运算、移位、计数、传送等功能部件和门电路）之间的联系。

在设计数据通路时，不但要考虑数据传送要求，还要考虑实现方案的合理性和经济性，争取共用或借用数据路径，尽量减少直接互连线的数量。通过加法器或内部总线实现寄存器间数据的传送可节省设备，但却失去了专用路径的操作并行性。

由加法器或总线建立的公共路径，1 次只能供给 1 对寄存器使用。如果同时有两对或两对以上的数据单元要求通过它传送数据时，需要在时间上错开。若在时间上无法错开，或为了提高操作的并行性，寄存器之间可设立专用路径。

典型的一地址指令计算机的数据通路如图 1 所示。围绕算术逻辑部件 ALU 把数据路径分为 3 组：

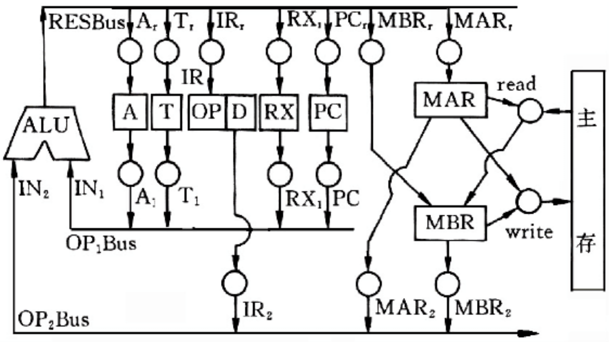


图 1　典型的一地址计算机数据通路

（1）由第一操作数总线 OP_1 Bus 供给 ALU 1 个操作数 IN_1　它来自累加寄存器 A 或中间结果寄存器 T 或指令计数器 PC 或变址寄存器 RX，它们分别由信号 A_1，T_1，PC_1，RX_1 控制。

（2）由第二操作数总线 OP_2 Bus 供给 ALU 另一个操作数 IN_2　它来自指令寄存器 IR 或存储器地址寄存器 MAR 或存储器缓冲寄存器 MBR，它们分别由信号 IR_2，MAR_2，MBR_2 控制。

（3）结果总线 RES Bus　它可把 ALU 运算结果送给上列各寄存器，控制信号分别为 A_r，T_r，PC_r，RX_r，IR_r，MAR_r 和 MBR_r。

ALU 输入数据的分组是按照指令要求进行划分的。ALU 完成的具体操作由操作码和时序信号配合进行控制。

例如取指令周期的操作：指令计数器 PC 在 PC_1 控制下，将指令地址送入 OP_1 Bus，通过 ALU 的输出，进入 RES Bus，在 MAR_r 信号控制下送入存储器地址寄存器 MAR。在读命令 Read 控制下，由主存读出 MAR 指定单元内容（指令）送入 MBR，在 MBR_2 信号控制下将 MBR 的内容经 OP_2 Bus 送入 ALU，经 RES Bus 在 IR_r 信号控制下将该信息送入指令寄存器 IR。

参考文献

金兰,金波. 计算机组织:原理、分析与设计. 北京:清华大学出版社,2006 (谢树煜)

shuju tongxin

数据通信(data communication) 指发送端与接收端交换和传输信息(通常为计算机数据)的过程和技术。数据通信可以狭义的理解为**数据传输**(data transmission)。 (张凌)

shuju tongxin jiekou

数据通信接口(data communication interface) **数据终端设备**(DTE)和**数据电路设备**(DCE)之间交换数据和控制信息的接口。数据通信接口的关键在于标准化,美国电子工业协会(EIA)发布了一系列标准化接口如 RS232C、RS423A、RS422A 等,并被国际电信联盟(CCITT、ITU-T)接纳。数据通信接口通常包含机械、电气、功能和规程逻辑四方面的重要特性。

机械特性

机械特性规定了接口的物理尺寸、形状、引脚的数量排列等。

电气特性

电气特性定义了接口与线路上传输信号的电平、传输距离、输入阻抗、时钟信号等。电气特性主要分为三类,即非平衡型、新的非平衡型和新的平衡型。

非平衡型的信号发送器和接收器均采用非平衡型方式工作,每个信号用一根导线传输,所有信号共用一根地线。定义 +5 v ~ +15 v 范围内的电平为二进制“0”,−5 v ~ −15 v 电平范围代表二进制“1”。非平衡型信号传输速率在 20 kb/s 以内,电线长度受限于 15 m,传输过程中信号易受干扰。

新的非平衡型标准中,发送器工作在非平衡型方式,接收器工作在平衡方式(即差分接收器)。每个信号用一根导线传输,而有两根不同方向的地线由所有信号共用。新的非平衡型标准用 +4 v ~ +6 v 电平表示“0”,−4 v ~ −6 v 电平表示“1”。信号传输速率与传输距离成反比,当传输距离达到 1000 m 时,信号传输速率在 3 kb/s 以下,在 10 m 以下的近距传输时,传输速率可达 300 kb/s。由于每个方向的传输独立使用地线,因此减少了线间干扰和外部干扰。

新的平衡型标准中,发送器和接收器均以差分方式工作,每个信号用两根导线传输,信号电平以两根导线上电平的差值表示。相对于某条导线,差值在 +4 v ~ +6 v 表示二进制“0”,差值在 −4 v ~ −6 v 为“1”。传输距离在 1000 m 时,传输速率小于 100 kb/s,但当进行 10 m 以内的近距传输时,速率可达 10 Mb/s。新的平衡性标准由于对每个信号采用双线传输,具有较好的抗干扰性。

功能特性

功能特性对接口引脚进行了功能定义,包括有数据引脚、控制引脚、时序引脚、接地引脚等。

规程逻辑特性

规程逻辑特性定义了实现特定应用时各种电路的执行顺序。

RS-232 是 EIA 发表的 OSI 基本参考模型物理层部分的数据通信接口标准,定义了连接器形状等物理特性、0 和 1 表示的电气特性以及信号意义的逻辑特性。RS-232C 是 RS-232 系列中较为广泛使用的标准,所用的连接器为 25 引脚,也称为 25 引脚 D-SUB。数据终端连接 RS-232C 连接器的公插头,通信设备连接母插座。RS-232C 电路工作在非平衡型方式,RS-232C 标准的连接器可以在 200 kb/s 以下的任何速率进行数据传输。RS-232C 标准对连接器 25 引脚的定义如图 1 所示。

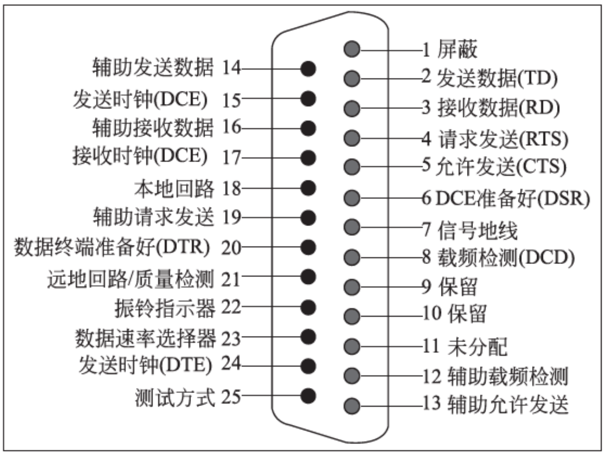


图 1 RS-232C 标准连接器引脚定义

参考文献

1. Forouzan B A. 数据通信与网络. 2 版. 吴时霖, 周正康, 吴永辉, 等译. 北京: 机械工业出版社, 2002

2. Gilbert Held. 数据通信. 6 版. 戴志涛, 卞佳

丽，郑岩，译. 北京：人民邮电出版社，2000

（王昊翔　张凌）

shuju tongxin shebei

数据通信设备（data communication equipment）　指使用各种通信介质进行数据传输的设备。数据通信设备的名词使用主要源于传统的电信行业（telecommunications industry），通信设备主要由专用硬件与计算机软件组成。包括各种数据终端、调制解调器、帧中继设备（FR）、DDN 设备、综合业务数字网（ISDN）、ATM 交换机、以太网交换设备、IP 路由设备等。在现代计算机通信网络中，数据终端一般是计算机或各种嵌入式设备。

传统电信行业分工相对清晰，分为最终用户（如企业、家庭、个人等）、电信运营商和通信设备制造商三大类。由最终用户自行选择和采购的数据通信设备称为**数据终端设备**（data terminal equipment，DTE），例如计算机、银行 ATM 终端等；安装在用户端（customer site）由电信运营商提供、电信运营商认证或电信运营商运行的通信设备称为**数据电路设备**（data circuit-terminating equipment，DCE），例如各种介质的调制解调器；数据电路设备 DCE 接入电信运营商运行的数据传输网络；数据电路设备 DCE 与数据终端设备之间的接口，称为**数据通信接口**。

参考文献

1. Forouzan B A. 数据通信与网络. 2 版. 吴时霖，周正康，吴永辉，等译. 北京：机械工业出版社，2002

2. Gilbert Held. 数据通信. 6 版. 戴志涛，卞佳丽，郑岩，译. 北京：人民邮电出版社，2000

（王昊翔　张凌）

shuju wajue

数据挖掘（data mining）　指从大量的、不完全的、有噪声的、模糊的、随机的原始数据中，提取隐含的、人们事先未知的、但又潜在有用的信息和知识的非平凡过程，也称数据中的知识发现（knowledge discovery in data，KDD）。它是一门涉及面很广的交叉学科，包括**计算智能**、**机器学习**、**模式识别**、信息检索、数理统计、**数据库**等相关技术，在商务管理、生产控制、市场分析、科学探索等许多领域具有广泛的应用价值。

数据挖掘过程粗略地可分为：数据准备、数据挖掘以及结果的解释和评估。其中，数据准备包括数据清理、数据集成、数据选择、数据变换等步骤；结果的解释和评估包括模式评估、**知识表示**及可视化等步骤；但研究的核心主要集中在第二步即数据挖掘方法上。

传统的数据挖掘按照其处理的任务不同可分为：分类、预测、聚类、离群点检测、演变分析、相关性分析、关联规则挖掘等。其中，分类、预测、聚类分析、离群点检测及演变分析又可归入机器学习的研究范畴；相关性分析可归入数理统计的研究范畴；只有关联规则挖掘是数据挖掘的特有任务。

设集合 $I=\{i_1,i_2,\cdots,i_n\}$，其中 i_k 是项，$k=1,2,\cdots,n$。与任务相关的数据库 D 是事务的集合，其中每个事务 T 是项的集合，使得 $T\subseteq I$。设 X 是项集，事务 T 包含 X，当且仅当 $X\subseteq T$。关联规则是数据库中蕴涵的项之间的有趣模式，形如：$X\rightarrow Y[s,c]$。其中 X,Y 是项集 I 的非空子集且 $X\cap Y=\varnothing$；s 称为规则的支撑度，是事务数据库中同时包含 X 和 Y 的事务在数据库中所占的比例，即联合概率 $Pr(XY)$；c 称为规则的置信度，是事务数据库中包含 X 的事务同时也包含 Y 的百分比，即条件概率 $Pr(Y|X)$；支撑度和置信度是用于描述规则兴趣度的两个度量。

例如：设 buys（x，diapers）→buys（x，beers）[50%，60%]是某超市销售数据库中蕴涵的一条关联规则，它表示买尿布的顾客很有可能同时买啤酒。这种类型的规则可用于发现顾客的购物模式，指导超市的布局和货物的分配等。

关联规则挖掘就是要找出数据库中蕴涵的所有同时满足用户给定的最小支撑度阈值（min_sup）和最小置信度阈值（min_conf）的关联规则。目前大多数关联规则挖掘方法采用两步走策略：①找到所有的频繁项集。所谓频繁项集（又称为大项集或频繁模式）是指那些支撑度大于最小支撑度阈值的项集；②由找到的频繁项集产生关联规则。即在找到的频繁项集中寻找满足最小置信度阈值的规则。

算法的性能主要由第一步决定，第二步在第一步的基础上很容易得到。因此，人们的研究工作主要集中在频繁模型发现任务上；而许多实际背景下的频繁模式具有特定的意义，如生物序列中的频繁模式可能是某个功能单元。因此，频繁模式挖掘比关联规则挖掘更具有一般性。

此外，数据挖掘不同于计算智能、机器学习、模式识别、数理统计、数据库等领域的主要特征是：具有海量数据处理的能力，面向特定的数据类型和数

据处理任务；并且目前数据挖掘研究的主要任务已由传统关系型数据的挖掘过渡到复杂类型数据（时序数据、文本数据、Web数据、数据流、网络数据、多源异构数据等）的挖掘，而Web数据挖掘也可归于信息检索的研究范畴。因此，我们将数据挖掘简要分为：**频繁模式挖掘**（Frequent Pattern Mining）、**序列挖掘**（Sequence Mining）、**数据流挖掘**（Data Stream Mining）、**文本挖掘**（Text Mining）、**Web挖掘**（Web Mining）、**图挖掘**（Graph Mining）和**时空数据挖掘**（Temporal-spatial Mining）等。具体地：

（1）频繁模式挖掘是关联规则挖掘的关键步骤，在各种类型的数据（如事务数据、序列数据、流数据、网络数据等）中都隐含着具有特定意义的、频繁出现的模式，挖掘此类模型是数据挖掘的传统任务。

（2）序列挖掘是针对序列类型数据的数据处理，数据特点是数据或其属性具有有序性或时序性。

（3）数据流挖掘是针对数据流的数据挖掘，数据特点是数据随时间变化快且数据量大。

（4）文本挖掘是针对文本类型数据的数据处理，数据特点是无结构或半结构且数据维度高。

（5）Web挖掘是针对互联网Web类型数据进行处理，数据特点是异构性强且具有无结构或半结构特征。

（6）图挖掘是针对社会网络数据、生物网络数据等复杂系统的图状或网状数据进行处理，数据特点是数据量大、具有较强的拓扑结构且复杂度高。

（7）时空数据挖掘是针对地图、遥感图像、交通控制、航线等具有时间或空间拓扑信息的数据进行处理，数据特点是数据间的时间或空间位置之间的相关度高。

目前，数据挖掘正由传统的单一数据的处理任务，向着复杂类型数据挖掘和多模数据（文本、视频、音频、网络拓扑等）挖掘的方向发展。相应地在方法上，除了以构建健壮有效挖掘模型为目标的模型驱动（model-driven）的研究方法外，从来自不同结构和模态的数据出发、以更好地解决实际问题为目标的数据驱动（data-driven）的研究方法日益受到重视，并已在信息检索、图像理解、多媒体内容分析等领域取得不少成功的应用。另外，动态数据的演化模式挖掘和社会媒体计算日渐成为当今数据挖掘领域的重要研究方向，**概率图模型**日益受到关注并有可能成为复杂类型数据挖掘的一个理论基础和重要工具。

参考文献

1. 王珊，等. 数据仓库技术与联机分析处理. 北京：科学出版社，1998

2. Han J W, Kamber M. Data mining: concepts and techniques. 3rd ed. Morgan Kaufmann, 2011

3. Hand D, Mannila H, Smyth P. Principles of data mining, The MIT Press, 2001

4. Rajaraman A, Ullman J D. Mining of massive datasets. Cambridge University Press, 2010

（贾彩燕　黄厚宽）

shuju wanzhengxing

数据完整性（data integrity）　**数据库系统**的一种特性，它能够保证数据库中的数据准确地反映它所表示的现实世界。通常，数据完整性是通过实施一系列完整性约束或完整性规则加以实现的。**关系数据库**的数据完整性可以分成**实体完整性**、**域完整性**和**参照完整性**三种类型。

实体完整性（entity integrity）　与主键的概念有关，它要求每个关系必须具有自己的主键，而且在这个关系中每个元组的主键的值是唯一的并且不能为空。例如，学生（学号，姓名，性别，生日，所在系）这一关系中，属性"学号"为主键。这样，当在学生关系中插入或修改元组时，系统就会根据实体完整性的规则，确保每个学生的学号必须具有值。

域完整性（domain integrity）　要求关系中的每个属性的取值必须有一个明确的范围，即值域。值域是具有同一类型的数据值的一个集合，关系表格的每列的取值必须来自该列所对应的值域。例如，前面所给的学生关系中，我们可以把"性别"属性的取值范围定义在{男，女}这个集合内，把"所在系"的值域定义成由一个大学所有的下属系组成的集合。这样，当在学生关系中插入或修改元组时，系统就会根据域完整性的规则，确保每个学生的性别和所在系属于事先定义的值域范围。

参照完整性（referential integrity）　与外键的概念有关，它要求一个关系的外键的值只能有两种状态：非空值和空值（null）。外键在通常情况下取非空值，这时该值是数据库中另一个关系里某个唯一元组的主键值。换句话说，非空的外键值直接指向了在另一关系中存在的唯一的一个元组。例如，选课（学号，课号，成绩，开课学期）是记录学生选课及成绩的关系，其中的"学号"属性是外键，因为它是前面所给的学生关系的主键。当一个选课元组的学号

不为空时，它代表了学生关系中某位唯一的学生，从而可以用这个外键值直接访问那条学生记录。因此，参照完整性表达了具有外键值的元组和这个唯一元组之间的一种特殊关系。和具体应用有关，少数情况下外键的值可以为空值。这时它表示在前面所说的两个元组之间不存在那种特殊关系，或者是那种特殊的元组关系尚不清楚。参照完整性的维护比实体完整性和域完整性都要复杂，因为它涉及了两个关系之间的多条记录。例如，当插入一条选课元组时，必须保证该元组“学号”的外键值指向了在学生关系中存在的一条记录。否则，插入将被拒绝，因为它违背了参照完整性的规则。同理，当删除一条学生元组时，必须保证选课关系中不再存在任何元组其外键指向该被删除的元组。

参考文献

1. 王珊，萨师煊. 数据库系统概论. 4 版. 北京：高等教育出版社，2006

2. Silberschatz A, et al. Database system concepts. 4th ed. McGraw-Hill Inc., 2002 （周立柱）

shuju yilai

数据依赖（data dependency） **关系数据库**中数据间的一种语义约束。它是通过元组属性值的相等与否来确定元组属性的相互联系。数据依赖的种类较多，主要的有**函数依赖**和**多值依赖**。

函数依赖 设 R(U)是定义在属性集 U 上的任一关系，X,Y 为 U 的子集。若 R 的任一实例 r 中的任意两个元组 s,t 在属性 X 上取值相等，即 $s[X]=t[X]$；则 s, t 在属性 Y 上取值也相等，即 $s[Y]=t[Y]$，则称属性 X 函数决定属性 Y，或属性 Y 函数依赖于 X；记为 $X\to Y$。

若 $X\supseteq Y$，则显然 $X\to Y$ 成立，称它为平凡函数依赖；若 $X\to Y$ 成立，且 X 中不存在真子集 X' 使得 $X'\to Y$ 成立，则称 $X\to Y$ 为完全函数依赖，否则称为部分函数依赖。

例：关系模式 R(C,S,Z)，C 为城市名，S 为街道名，Z 为邮政编码，则函数依赖 $Z\to C$，$CS\to Z$ 成立。

多值依赖 设 R(U)是定义在属性集 U 上的关系模式，r 是 R(U)上的任意关系实例，X,Y 为 U 的子集。若 r 中的任意两个元组 t_1,t_2，在属性 X 上取值相等 $t_1[X]=t_2[X]$，则在 r 中存在元组 t_3,t_4 取值为

$t_3[X]=t_1[X]$ $t_3[Y]=t_1[Y]$

$t_3[U\text{-}X\text{-}Y]=t_2[U\text{-}X\text{-}Y]$；

$t_4[X]=t_2[X]$ $t_4[Y]=t_2[Y]$

$t_4[U\text{-}X\text{-}Y]=t_1[U\text{-}X\text{-}Y]$

称属性 X 多值决定属性 Y，或属性 Y 多值依赖于 X；记为 $X\to\to Y$。若 $U=X\cup Y$，则 $X\to\to Y$ 成立，称它为平凡多值依赖。

数据依赖在关系数据库中广泛存在，这可能引起数据冗余，并导致关系数据库上的修改、插入和删除操作产生异常现象。数据依赖是关系数据库的重要研究内容，关系数据库的设计，特别是规范化理论是建立在它的基础上的。

参考文献

Abiteboul S, Hull R, Vianu V. Foundations of databases. Addison-Wesley, 1995

（谈子敬 施伯乐）

shuju yinsi

数据隐私（data privacy） 用来描述个人控制其不愿他人知道或他人不便知道的个人数据的能力。数据隐私范围很广，涉及数据管理中的数据收集、数据存储、数据处理和数据发布等各个阶段。

在数据收集阶段，数据收集者可能获取到个人的隐私信息。为了减少信息收集过程中对隐私权的破坏，目前出现了数据收集者需要遵守的隐私数据收集协议。例如，万维网联盟（W3C）公布了隐私保护协议 P3P（platform for privacy preferences），这一协议要求在个人访问 Web 站点过程中，个人应被告知该站点所收集的数据类型、数据使用途径和方式等。这些协议增强了个人在数据收集阶段中隐私保护的主动性。

在数据存储阶段，应避免其他非授权的用户访问个人的隐私数据。通常可以使用数据库安全技术实现这一阶段的隐私保护，如可以使用访问控制、数据加密等。数据的访问控制标准包括自主访问控制、强制访问控制和基于角色的访问控制等。数据加密使用已有的密码技术和算法进一步保护存储的隐私数据。

在数据处理阶段，需要考虑数据推理带来的隐私数据泄露。非授权用户可能通过分析多次查询的结果，或者基于完整性约束信息，推导出其他用户的隐私数据。此外，还需要考虑**数据挖掘**对隐私保护的影响。数据挖掘的目的是获取先前未知的模式或信息，而这些挖掘结果中可能包含用户的隐私数据。因此，一般在数据挖掘之前对包含隐私数据的原始数据进行变换、扰乱等操作，减少和隐私数据相关的

挖掘结果。

在数据发布阶段,应使包含隐私的数据发布结果满足特定的安全性标准。以关系数据表为例,发布的数据表首先不能包含原有表的候选码,同时还要考虑准标识符的影响。准标识符是能够唯一确定大部分记录的属性集合。在现有安全性标准中,*k*-匿名化(*k*-anonymization)标准要求每个具有相同准标识符的记录组中至少包括 *k* 条记录,从而控制攻击者判别隐私数据所属个体的概率。l-多样化标准(l-diversity)较 *k*-匿名化严格,限制了在每个具有相同准标识符的记录组中,敏感属性值的分布满足和 l 相关的条件,如至少包含 l 个不同的值。*t*-临近标准(*t*-closeness)更加严格,要求在每个具有相同准标识符的记录组中,敏感属性值的分布尽量接近该属性值的全局分布,从而保证攻击者不能从发布数据中推导出额外的隐私数据。

参考文献

Liu L, Ozsu M T. Encyclopedia of database systems. Springer, 2009

（杨冬青　高军）

shuju zhiliang

数据质量(data quality)　针对应用需要,数据在完备性、正确性、一致性和时效性等方面所处的状态。由于观测仪器的误差、精度限制、人为因素、数据集成和数据模式的变更等多种原因,数据可能存在不正确和不完备等情况,这被称为数据质量问题。数据质量管理的首要问题是使用度量方法对数据中存在的问题给出评估。数据质量的度量方法通常由多个维度构成,对维度的选择和具体的应用相关。质量评估的维度用于描述潜在的数据质量问题,因此可以根据其性质进行分类。常见和重要的维度包括:

完备性　缺失和不完整的数据是很多应用中最主要的数据质量问题,通常可以使用空值在所有值中所占的比例来衡量。

正确性　数据质量问题经常由于度量和观测误差引起。正确性被定义为无错误和已经获得相关验证的程度。

一致性　作为数据质量的度量,一致性被定义为数据满足给定的约束或商业规则的程度。这些规则可以是数据库中所定义的完整性约束。

时效性　过时的数据往往会引起数据质量问题,时效性作为质量评估的维度之一,被用来描述现有数据在多大程度上是最新的。

参考文献

Batini C, Scannapieco M. Data quality-concepts, methodologies and techniques. Springer, 2006

（施伯乐　谈子敬）

shuju zhongxin

数据中心(data center)　指企业或者组织放置相关计算机业务系统及其相关部件并对业务系统和数据资源进行集中、集成、共享和分析的场所。数据中心也被称为服务器农场(server farm),它一般包含系统冗余和备用电源、冗余网络连接、环境控制(例如空调、灭火器)和安全设备。

一个数据中心占用一幢大楼的一个房间,一层或多层,甚至整栋大楼。大部分设备常常放在具有 19 英寸的隔层的机架中,这些机架成排放置,形成一个走廊,允许人们从前面或后面访问隔层。数据中心的物理环境,包括温度和湿度、冷却装置、电源供应、地面升高高度、防火系统、监视和预警系统等都有严格要求。

数据中心功能

随着信息技术的发展和应用的深入,数据中心功能的内涵已经发生了变化。其发展可分为四个阶段:数据存储中心阶段、数据处理中心阶段、数据应用中心阶段、数据运营服务中心阶段。

(1)数据存储中心阶段　在数据存储中心阶段,数据中心主要承担的功能是数据存储和管理。在信息化建设早期,用来作为办公自动化机房或电子文档的集中管理场所。此阶段的典型特征是:数据中心仅仅是便于数据的集中存放和管理;数据单向存储和应用;无专人的维护;关注新技术的应用。由于数据中心的功能比较单一,整体可用性也很低。

(2)数据处理中心阶段　在数据处理中心阶段,基于局域网的制造资源规划(MRP)、企业资源规划(ERP)以及其他的行业应用系统开始普遍应用,数据中心开始承担核心计算的功能。此阶段的典型特征是:面向核心计算;数据单项应用;机构开始组织专门的人员进行集中维护;开始关注计算的效率及对机构运营效率的提高。整体上可用性较低。

(3)数据中心应用阶段　随着广域网或**互联网**的应用开始普及,信息资源日益丰富,开始关注挖掘和利用信息资源。组件化技术及平台化技术广泛应用,数据中心承担着核心计算和核心业务运营支撑,

满足需求的变化成为数据中心的核心特征之一。这一阶段典型数据中心称作“信息中心”。此阶段的特征是：面向业务需求，数据中心提供可靠的业务支撑；数据中心提供单向的信息资源服务；对系统维护上升到管理的高度，从事后处理到事前预防；开始关注信息技术的绩效；数据中心要求较高的可用性。

（4）数据运营服务中心阶段　从现在技术发展趋势分析，基于互联网技术的组件化、平台化的技术将更加广泛地应用以及数据中心基础设施的智能化，使得组织运营实现高度自动化。数据中心将承担组织的核心运营支撑、信息资源服务、核心计算、数据存储和备份，并确保业务可持续性计划实施等。业务运营对数据中心的要求将不仅仅是支持，而是提供持续可靠的服务。在这个阶段，数据中心将演进成为机构的数据运营服务中心。

数据运营服务中心的含义包括以下几个方面：机构数据中心不仅管理和维护各种信息资源，而且运营信息资源，确保价值最大化；IT应用随需应变，系统更加柔性，与业务运营融合在一起，实时的互动，业务与信息技术密不可分；信息技术服务管理成为一种标准化的工作，并借助它实现集中的自动化管理；信息技术绩效成为信息技术服务管理工作的一部分；不仅仅关注信息技术服务的效率，信息技术服务质量成为关注重点；数据中心要求具有高可用性。

所谓“新一代数据中心”的定义，就是通过自动化、资源整合与管理、虚拟化、安全以及能源管理等新技术的采用，解决目前数据中心普遍存在的成本快速增加、资源管理日益复杂、信息安全等方面的严峻挑战以及能源危机等尖锐的问题，从而打造与行业/企业业务动态发展相适应的新一代企业基础设施。新一代数据中心所倡导的“节能、高效、简化管理”也已经成为众多数据中心建设时的参考标准。

（武永卫）

shuju zhongxin guanli

数据中心管理（data center management）

为确保数据中心数据信息安全、系统运行稳定，履行数据集中处理、信息技术系统运行维护和技术保障等职责所需进行的管理活动。数据中心管理工作分为基础支撑与服务运行两大类。

基础支撑管理　为保障数据中心日常运行维护、服务运行，对数据中心的基础环境及技术保障的管理。包括以下几方面：

（1）连续性管理　目的是在灾难发生之后，数据中心基础设施和服务能够在最短时间内得到恢复，以支持核心业务。连续性管理的主要活动包括：确定信息技术服务连续性管理的范围、业务影响度分析、风险评估、确定连续性策略、组织和实施规划、预防措施和恢复方案等。

（2）容量管理　指为了确保数据中心在正常、异常、紧急等不同状况下，通过合理的协调资源，提供既定的服务以满足业务要求。

（3）信息安全管理　为确保信息资产不易遭受已知风险的侵袭，尽可能规避未知风险，通过保证信息资产的保密性、完整性和可用性，以保证信息系统稳定、可靠和安全的运行，确保各项业务顺利开展。

（4）配置管理　通过监控组件的运行状态，核实、控制基础设施中实施的变更以及配置项之间的关系，以确保能够准确反映现存配置项的实际状况。配置管理高度依赖变更和发布管理，同时也为其他活动提供准确的信息和文档。

服务运行管理　为实现对各类用户的服务承诺，为用户提供良好服务所采取措施的相应管理活动。包括服务级别管理、服务报告管理、可用性管理、业务关系管理、事件管理、问题管理以及变更管理等方面。

（1）服务级别管理　为实现承诺的数据中心服务和提高服务质量，通过定义、谈判、评价并管理服务级别，确保服务能够得到及时更新和持续有效。

（2）服务报告管理　通过及时、可靠、准确的报告以及有效的沟通，为经营管理决策提供依据，以确保服务报告满足特定要求和用户需求。

（3）可用性管理　确保实现服务协议承诺的一种过程控制，其目的是要提供符合预定可用性级别且成本合理的服务，以帮助企业实现业务目标。

（4）业务关系管理　基于对用户及其业务的了解，通过建立沟通渠道与窗口，及时获取与跟踪用户对服务的反馈等形式，建立并维护服务提供者与用户之间的良好关系。

（5）事件管理　在对用户或企业正常的业务活动带来最小影响的情况下，尽快恢复到原服务级别，尽可能减少事件对业务的影响。事件管理应遵循以下几个主要原则：事件要快速解决、尽快恢复服务，对事件要有清晰的定义和分类，对事件的记录要完整、有效。

（6）问题管理　指以“探求事件根源，寻找彻

底解决办法,预防为主"的原则,采取措施消除引起事件的深层次根源,以防止事件的再次发生,包括主动性问题管理和被动性问题管理两类活动。被动性问题管理找出导致以前的事件发生的根本原因,提出解决措施或纠正建议;而主动性问题管理是通过找出基础设施中的薄弱环节来阻止事件的再次发生。

(7) 变更管理　以控制变更为目的,通过记录、分类和过滤变更,确定变更的优先级,组织变更评审会议,评估变更的影响及所需的资源,审批变更,提供管理报告及其他相关信息等主要活动,以防范变更风险,确保变更有序进行。

数据中心管理的发展趋势　环保及节能是能耗巨大的数据中心面临的首要问题,为此,必须对数据中心的能耗进行全方位、全生命周期管理,从规划、设计、施工直至运维各个阶段始终树立节能意识,以提高数据中心的能源利用率。新一代数据中心还必须适应云计算环境,即根据实际应用状况进行虚拟动态调整。**云计算**的广泛应用及数据中心业务不断扩展要求提供的设施具备动态扩展功能。采用模块化建设是近来兴起的数据中心建设方法,即其基础设施是由多个具备独立功能及统一接口的模块以搭积木的方式组成,从而可按需灵活配置降低初期的投入,方便扩充和运行维护。新一代数据中心的总控设施将整合系统管理平台、流程管理平台及环境设施管理平台,形成一体化的运行管理模式。

(华建兴　苏振泽)

shuju zhongduan shebei

数据终端设备(data terminal equipment, DTE)　在数据通信设备中属于用户终端的一类设备。例如早期计算机外部设备中的字符型哑终端、计算机图形显示终端,计算机通信系统中的个人计算机、服务器等。

对于局部使用环境,数据终端设备 DTE 可以直接连接,例如绘图仪与打印机可以在本地直接连接计算机主机。

对于远程使用环境,数据终端设备 DTE 通常需要通过**数据电路设备** DCE 接入通信传输线路或通信网络进行远距离数据传输。此时,数据终端设备 DTE 与数据电路设备 DCE 之间必须采用标准**数据通信接口**,常用的接口标准有 RS232C、RS422、RS449、V. 24、V. 35、X. 21 等。

参考文献

1. Behrouz A. Forouzan. 数据通信与网络. 2版. 吴时霖、周正康、吴永辉,等译,北京:机械工业出版社,2002

2. Gilbert Held. 数据通信,6 版. 戴志涛,卞佳丽,郑岩,译. 北京:人民邮电出版社,2000

(王昊翔　张凌)

shuli tongji

数理统计(mathematical statistics)　以概率论为基础的一门数学分科。对所研究的随机现象进行多次观测或试验(称为取样),研究如何合理地安排试验以获取数据资料(称为试验设计),并根据所得的数据推断所研究对象的统计特性(称为统计推断或预测),直至为采取一定的决策或行动提供依据和建议的学科。

用数理统计方法解决实际问题时,一般步骤为:建立模型、收集或整理数据、统计推断、预测和决策。

所谓建立模型,是指对所研究问题的总体服从某种分布作出假定。这要依赖于对所研究现象的专业知识、历史经验以及概率论的知识。比如在研究电子元件(经过老化试验)的寿命时,一般假定它们服从指数分布,而当考虑射击误差时,即可认为落点服从高斯分布。一般情况下,则可将测量数据描绘在各种分布的概率纸上,并与某一分布曲线比较,如果接近就认为总体服从这种分布,这就是数理统计中的非参数检验。

数据的收集一般有普查、抽查和设计专门试验三种方式。普查的结果没有随机性,不属于数理统计研究的范围,现实中许多数据是不能普查的,比如产品的寿命,人们不能够为取得数据而把产品消耗光。数理统计关心的是抽查,也就是从产品中抽取一部分,观察其某种指标,从而推断全部产品的性质。关于如何进行抽样的研究构成了数理统计的一个分支,即抽样论。为了某一目的,专门安排试验去获得数据,这在数理统计中也是常见的。比如为了研究农作物的产量与肥、水、光照、土壤的关系,可以专门设计各因素在特定的各种量下的试验,正交试验法就是这样一种优化的试验设计方法。

有了数据后,再把它们用适当的方式制成图表,比如样本的直方图,由此可以看出一些粗略的规律性,并计算其样本均值、样本方差,以供进一步做统计推断。

统计推断包括参数估计、假设检验、回归分析、

方差分析等内容,这些是数理统计的主题,在此基础上进行统计预测,比如用回归方程来预测量的变化。统计决策是在统计推断的基础上,对人们决策的一种建议。比如一个商店应该考虑商品存储的最佳数量,太少,供不应求,对商店是一个损失;太多,造成积压,不但存储本身需要费用,而且由于存储过长造成的损坏及资金的积压都是一种损失。我们可以把带有随机性的诸多因素综合在一起,考虑商店的经济效益。统计决策可以帮助我们得出各种商品的最佳存储量,这方面的研究已经发展成为储存论,它是统计决策的一部分。下面列出数理统计中经常要遇到的最基本概念。

母体(或总体) 指研究对象的全体。比如欲研究某灯泡厂生产灯泡的使用寿命,则把该生产厂在相同生产条件下生产的灯泡寿命全体看成是一个母体,它是一个随机变量。

样本(或子样) 对母体作 n 次独立的观察,得到 n 个随机变量 $X_1, X_2, \cdots, X_n$,它们相互独立,且与母体 X 同分布。称如此的 X 是一个样本容量为 n 的样本(或子样)。仍以灯泡的寿命为例,如取 $n=3$,即任取 3 个灯泡作寿命试验,得到三个数 (x_1, x_2, x_3),这三个数就是样本 (X_1, X_2, X_3) 的一次观察值,而 (X_1, X_2, X_3) 本身就是三元的随机变量。

统计量 设 $(X_1, X_2, \cdots, X_n)$ 为来自母体 X 的子样,h 为 n 元变量的已知函数,如果 $T = h(X_1, X_2, \cdots, X_n)$ 仍是一个随机变量,则称 T 为统计量。一般地,h 是连续或分段连续的已知函数,此时 T 必为统计量。

χ^2 分布 若 $\xi_1, \xi_2, \cdots, \xi_n$ 为 n 个相互独立且服从 $N(0,1)$ 分布的变量,则 $\chi^2 = \sum_{i=1}^{n} \xi_i^2$ 的密度函数为

$$f_{\chi^2}(x) = \begin{cases} \dfrac{1}{2^{\frac{n}{2}}\Gamma\left(\dfrac{n}{2}\right)} x^{\frac{n}{2}-1} e^{-\frac{x}{2}}, & x \geqslant 0 \\ 0, & \text{其他} \end{cases}$$

此时称随机变量 χ^2 服从自由度为 n 的 χ^2 分布,简记为 χ_n^2。

设母体 $X \sim N(\mu, \sigma^2)$,子样 $(X_1, X_2, \cdots, X_n)$ 来自母体 X,子样均值和子样方差分别为

$$\overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i, \quad s^2 = \frac{1}{n}\sum_{i=1}^{n}(X_i - \overline{X})^2$$

则可知 $\overline{X} \sim N\left(\mu, \dfrac{\sigma^2}{n}\right)$, $\dfrac{ns^2}{\sigma^2} \sim \chi_{n-1}^2$,且 $\overline{X}$ 与 s^2 相互独立。

t 分布 设 ξ, Z 相互独立,$\xi \sim N(0,1)$,$Z \sim \chi_n^2$,则 $t = \xi \Big/ \sqrt{\dfrac{Z}{n}}$ 服从自由度为 n 的 t 分布,简记为 t_n,其密度函数为

$$f_t(x) = \frac{\Gamma\left(\dfrac{n+1}{2}\right)}{\sqrt{n\pi}\,\Gamma\left(\dfrac{n}{2}\right)}\left(1 + \frac{x^2}{n}\right)^{-\frac{n+1}{2}}$$

F 分布 设 $\xi \sim \chi_m^2, \eta \sim \chi_n^2$ 且 ξ 与 η 相互独立,则

$$\zeta = \frac{\xi / m}{\eta / n}$$

服从自由度为 m 及 n 的 F 分布,简记为 $F(m,n)$,其密度函数

$$f_\zeta(x) = \begin{cases} 0, & x \leqslant 0 \\ \dfrac{\Gamma\left(\dfrac{m+n}{2}\right)}{\Gamma\left(\dfrac{m}{2}\right)\Gamma\left(\dfrac{n}{2}\right)} m^{\frac{m}{2}} n^{\frac{n}{2}} \dfrac{x^{\frac{m}{2}-1}}{(mx+n)^{\frac{m+n}{2}}}, & x > 0 \end{cases}$$

上述 χ^2 分布、t 分布、F 分布在统计中应用得很多,在统计用表中给出了这些分布的临界值表,供大家使用。

参考文献

1. 王梓坤. 概率论基础及其应用. 北京:科学出版社,1976

2. 陈希孺. 数理统计引论. 北京:科学出版社,1981

(金治明)

shumo/moshu zhuanhuanqi

数模/模数转换器(digital to analog/analog to digital converter) 实现数字信号与模拟信号间相互转换的集成电路。其中,把数字信号转换成模拟信号的电路是数模转换器,简称 D/A 或 DAC;把模拟信号转换成数字信号的电路是模数转换器,简称 A/D 或 ADC。因为 D/A、A/D 互为逆过程,有时也称 A/D 为编码器电路,D/A 为译码器电路。

为了实现数字信号和模拟信号之间的互相转换,无论是 A/D 还是 D/A,其电路结构中都既有数字电路又有模拟电路,是名副其实的数模混合电路。因为计算机是数字系统,外部世界是模拟系统,所以 A/D 主要用在计算机的输入端,D/A 主要用在计算机的输出端。正如计算机的输入和输出经常密不可分一样,大部分的嵌入式控制系统都既包含 A/D 又包含 D/A。

实现数模转换的基本思路是先把输入数字信号的每个二进位分别转换为一个模拟量(电流或电压),其幅值正比于该二进位的权值。再把这些分离的模拟量叠加在一起,就可得到与输入数字量对应的输出模拟量。D/A 转换器电路的通道框图如图 1 所示。D/A 按开关电路类型可分为电流型和电压型;按网络阵列可分为权电阻型和梯形电阻网络型。

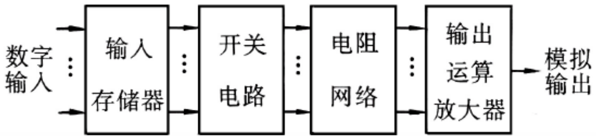


图 1　D/A 转换器电路通道框图

图 2 给出的是梯形电阻网络型 4 位 D/A 电路示意图,其中使用的电阻仅有 R 和 2R 两种阻值。在这种结构中,电流是转换媒介,开关电路与电阻网络直接对应。输入的 4 个二进位直接决定对应电子开关的状态:输入“1”时开关闭合,支路中有电流通过;输入“0”时开关断开,支路中没有电流通过。分析电阻网络的结构可知,流经各个电子开关的电流正比于该开关对应的二进位的权值。位于图中下方

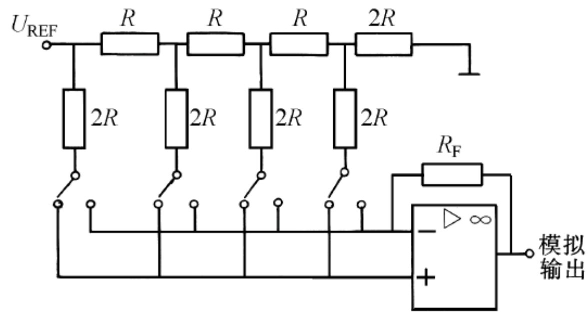


图 2　梯形电阻网络型 D/A 转换器电路

最左侧的输入端是输入数据的最高位,最右侧的为输入数据的最低位。与各二进位对应的支路上的电流在运算放大器输入端汇总后,形成最终的模拟电压输出结果。

实现模数转换的基本思路是把输入模拟信号的特性值或幅值与一组标准的参考模拟信号进行比较,再根据比较结果,将信号转换成由若干二进位表示的数字信号。A/D 转换器电路的通道框图如图 3 所示,其中前置滤波器用来消除信号中的噪声,取样保持电路使输入信号在转换期间保持稳定。A/D 按转换技术可分为直接转换型和间接转换型,其典型工作方式有比较式和积分式两种。

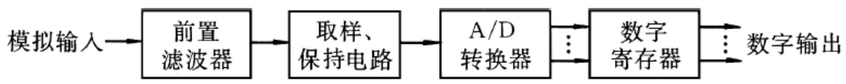


图 3　A/D 转换器电路通道框图

图 4 给出了一种比较式 A/D 转换器的结构示意图,它基于内置的 D/A 转换器,采用类似于折半查找的方法,得出与输入信号最接近的数字输出,也称为逐次逼近式 A/D 转换器。图中寄存器的宽度 n 等于输出数字信号的位宽,整个转换过程分为 n 步。第 1 步,先由控制逻辑置寄存器最高位为 1,其余位为 0;寄存器的输出经 D/A 变成模拟信号,与输入信号进行比较,若输入信号更小,则由控制逻辑置寄存器最高位为 0。第 2 步,保留寄存器最高位的内容不变,置次高位为 1;仍将寄存器输出经 D/A 变成的模拟信号与输入信号进行比较,并依同样的规则,确定次高位的置值。依次类推,直到最低位。这时,寄存器的输出,就是与输入信号幅度对应的二进制数值。

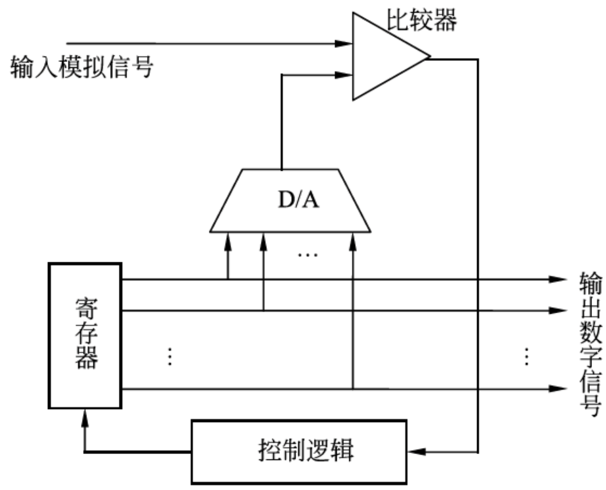


图 4　比较式 A/D 转换器框图

积分式 A/D 转换器采用间接转换方式,先用积分器对模拟输入电压作常数时间的积分(相当于用输入电压给电容器充常数时间的电),积分时间可用给计数器预设一个倒计时初值来控制。倒计时结束后,积分电路在刚才积分值的基础上,对标准电压进行反向积分,直至积分器输出为 0(相当于用标准电压给电容器放电)。计数器记录下来的反向积分时间,就是与输入电压幅值对应的二进制数值。

比较式 A/D 的特点是转换速度和精度都较高，其精度取决于计数器位数和参考电压精度。积分式 A/D 的特点是精度高、速度慢，其精度取决于参考电压和内部时钟频率的精度。

数模/模数转换器的主要特性参数包括转换速率、精度、温度系数、功耗等。随着微电子技术的发展，数模/模数转换器的基础工艺已从双极型电路过渡到互补金属-氧化物-半导体(CMOS)电路，并朝着高速度、高精度、高集成度方向发展。A/D 和 D/A 除了以独立的芯片产品应用于各类电子系统外，基于 CMOS 工艺实现 A/D 和 D/A 转换的**知识产权(IP)核**也大量应用在**片上系统**中。

参考文献

1. 张凡，盛珣华. 微机原理与接口技术. 北京：中国铁道出版社，2000

2. Holdsworth B. 数字逻辑设计. 4 版. 李仁发，等译. 北京：人民邮电出版社，2006

（孙育宁　唐志敏）

shuzhi bijin

数值逼近（numerical approximation）　关于如何使用容易数值计算的函数来近似地代替任意函数的方法与过程。

对于给定的较广泛的函数类 F 中的函数 f，从较小的子类 H 中寻求一个函数 h，使之在某种意义下近似代替函数 f，以便于 f 的计算与处理。函数逼近的主要内容有：对于某些特定的被逼近函数类 F 与逼近函数类 H，讨论逼近的可能性；最佳逼近的存在性、特征、唯一性、误差估计以及逼近算法等。它是现代数值分析的基本组成部分，除自身具有独立学科分支的意义外，还可用于构造数值积分、求函数零点、解微分方程和积分方程等的近似方法。

设被逼近函数 $f \in C[a,b]$，逼近函数类记作 $H \subset C[a,b]$，用

$$\|f-g\|_p = \left[\int_a^b |f(x)-g(x)|^p \omega(x)\,\mathrm{d}x\right]^{\frac{1}{p}},\quad 1 \leqslant p < +\infty$$

来度量函数 f 与 g 的距离，其中 $\omega(x) > 0$ 为选取的权函数。特别地，当 $p=\infty$ 时，通常取 $\omega(x) \equiv 1$，此时

$$\|f-g\|_\infty = \max_{a\leqslant x\leqslant b} |f(x)-g(x)|$$

这种度量 F 的逼近称为**一致逼近**；另一种重要情形是当 $p=2$ 时，称为**均方逼近**或平方逼近。

最佳逼近　若 $h \in H$ 满足

$$\|f-h\|_p = \inf_{g\in H} \|f-g\|_p$$

则称 h 为距离度量（1）意义下 f 在 H 中的最佳逼近。对于 $p=2$ 和 ∞ 时，相应的 h 分别称为 f 在 H 中的最佳平方逼近和最佳一致逼近，后一种情况又称切比雪夫逼近，它是由 П. Л. Чебышев 在 1854 年首先开始研究的。

多项式逼近　指 H 取作多项式类的情形。1885 年 Weierstrass 的著名定理给出了关于用多项式一致逼近连续函数到任意精度的可能性：若 $f \in C[a,b]$，则对任意 $\varepsilon > 0$，都存在代数多项式 p，使 $\|f-p\|_\infty < \varepsilon$。它同时给出了用三角多项式一致逼近周期连续函数到任意精度的结果。H. L. Lebesgue 曾证明了对于连续函数类而言，切比雪夫级数展开式的部分和是最佳一致多项式逼近的很好近似。

有理逼近　指 H 取作

$$R_m^n[a,b] = \left\{R = \frac{P}{Q} \,\middle|\, \partial P \leqslant n, \partial Q \leqslant m, Q > 0, \frac{P}{Q}\text{不可约}\right\}$$

的情形，其中 P,Q 为多项式函数，∂P，∂Q 表示它们的次数，当 $f \in C[a,b]$ 时，f 在 $R_m^n[a,b]$ 中的最佳一致逼近 R^* 存在且唯一的充要条件是：在 $[a,b]$ 上有一组点 $a \leqslant x_1 < x_2 < \cdots < x_k \leqslant b$ $(k \geqslant \max\{n+\partial Q, m+\partial P\}+2)$ 使得误差 R^*-f 在这些点上达到其最大绝对值，而且符号正负交替变化，即

$$R^*(x_i)-f(x_i) = (-1)^i\lambda,\quad i=1,2,\cdots,k$$

$$|\lambda| = \|R^*-f\|_\infty$$

一种特殊的有理逼近称为帕德逼近，它应用于数值分析的一些领域以及物理学和化学的某些计算问题中。设 $f(x) = \sum_{i=0}^{\infty} \alpha_i x^i$，若存在多项式

$$P_n(x) = \sum_{i=0}^{n} p_i x^i$$

$$Q_m(x) = \sum_{i=0}^{m} q_i x^i$$

满足条件 $f(x) - P_n(x)/Q_m(x) = O(x^{n+m+1})$ 且

P_n/Q_m 不可约，$Q_m(0)=q_0=1$，则称 $P_n(x)/Q_m(x)$ 为 $f(x)$ 在点 $x=0$ 处的 (n,m) 阶帕德逼近，并简记为 $[n/m]=P_n(x)/Q_m(x)$。若 $[n/m]$ 存在则必唯一。已证明帕德逼近是最佳局部有理切比雪夫逼近。帕德将 $[n/m]$ 依自然顺序排列成表格

$$\begin{array}{ccccc} [0/0] & [0/1] & [0/2] & [0/3] & \cdots \\ [1/0] & [1/1] & [1/2] & [1/3] & \cdots \\ [2/0] & [2/1] & [2/2] & [2/3] & \cdots \\ [3/0] & [3/1] & [3/2] & [3/3] & \cdots \\ \vdots & \vdots & \vdots & \vdots & \end{array}$$

称为帕德表，实践表明，帕德表中主对角线上及其两侧的元素 $[m/m]$ 和 $[n/n\pm1]$，在 $n+m$ 相等的诸元素中，通常有更好的逼近效果。

参考文献

1. Натансон И П. 函数构造论. 徐家福，等译. 北京：科学出版社，1965

2. Cheney E W. 逼近论导引. 徐献瑜，等译. 上海：上海科学技术出版社，1981

（沈毅）

shuzhi jifen

数值积分（numerical integration） 从近似计算角度出发，研究如何采用有效的数值计算过程求定积分近似值的方法与理论。通常求定积分

$$I=\int_a^b f(x)\mathrm{d}x \tag{1}$$

的方法是先找出被积函数的原函数 $F(x)$，然后由 $F(x)$ 求得积分(1)的值，即 $I=F(b)-F(a)$。然而实际情况是：有很多的积分不能解析地求解；有些积分虽然可以解析地求解，但原函数比被积函数复杂得多，且难于给出最后的数值结果；还有不少情况，被积函数 f 没有具体表达式，无法采用解析求解。因此，在实际计算时，常采用数值积分方法。

数值积分公式 一般形如

$$\int_a^b \rho(x)f(x)\mathrm{d}x \cong \sum_{i=0}^{n} A_i f(x_i) \tag{2}$$

的近似公式，为数值积分公式，也称求积公式。$\rho(x)$ 是 $[a,b]$ 上大于等于零的权函数，x_i 和 $A_i(i=0,1,2,\cdots,n)$ 分别称为求积结点和求积系数，式(2)右端称为求积和，差式

$$E(f)=\int_a^b \rho(x)f(x)\mathrm{d}x-\sum_{i=0}^{n} A_i f(x_i) \tag{3}$$

称为求积误差，区间 $[a,b]$ 为有限，也可为无限。由式(3)知，构造求积公式的问题就是确定 x_i 和 A_i，使 $E(f)$ 在某种意义下尽可能地小。

代数精度 若式(2)对 $f(x)=x^k(k=0,1,\cdots,d)$ 精确成立，即 $E(f)=0$，而当 $f(x)=x^{d+1}$ 时，式(2)不再是精确等式，则称式(2)的代数精度为 d。对一般连续函数 $f(x)$ 来说，d 越大，$E(f)$ 越小，因此，可以用代数精度高低说明求积公式的优劣。

牛顿–柯特斯求积公式 考虑权函数为 1 且 $[a,b]$ 为有限时的积分，可以通过插值途径来建立数值积分公式。将 $[a,b]$ 分为 n 等份，$h=(b-a)/n$，在 $n+1$ 个等距结点 $x_i=a+ih$ $(i=0,1,\cdots,n)$ 上，函数值为 $f(x_i)=y_i$ $(i=0,1,\cdots,n)$，由 $f(x)$ 的 n 次拉格朗日插值多项式而建立起的积分公式为

$$\int_a^b f(x)\mathrm{d}x$$

$$=\sum_{k=0}^{n} A_k f(x_k)+\frac{1}{(n+1)!}\int_a^b f^{(n+1)}(\xi)\prod_{j=0}^{n}(x-x_j)\mathrm{d}x$$

$$=I_n(f)+E_n(f) \tag{4}$$

其中

$$A_k=\int_{x_0}^{x_n}\prod_{\substack{j=0\\j\neq k}}^{n}\frac{x-x_j}{x_k-x_j}\mathrm{d}x,\quad k=0,1,\cdots,n;\ \xi\in[a,b]$$

且依赖于 x，$E_n(f)$ 为拉格朗日插值多项式余项。

若令 $x=a+th$ $(0\leqslant t\leqslant n)$，$\mathrm{d}x=h\mathrm{d}t$，$x_k=a+kh$，由式(4)得到

$$A_k=C_k^{(n)}(x_n-x_0)$$

其中

$$C_k^{(n)}=\frac{(-1)^{n-k}}{nk!(n-k)!}\int_0^n\prod_{\substack{j=0\\j\neq k}}^{n}(t-j)\mathrm{d}t$$

这时 $I_n(f)$ 和 $E_n(f)$ 分别为

$$I_n(f)=(x_n-x_0)\sum_{k=0}^{n} C_k^{(n)} f_k \tag{5}$$

$$E_n(f)=\int_a^b\frac{f^{(n+1)}(\xi)}{(n+1)!}\prod_{j=0}^{n}(x-x_j)\mathrm{d}x$$

$$=\frac{1}{(n+1)!}\int_a^b f^{(n+1)}(\xi)\prod_{j=0}^{n}(x-x_j)\mathrm{d}x \tag{6}$$

公式(5)称为牛顿–柯特斯公式，$C_k^{(n)}$ 称为柯特斯系数，其值依赖于 $[a,b]$ 等分数 n，其代数精度至少为 n。

当 $n=1$ 时，由公式(4)得到**梯形公式**及其误差

$$I_1 = \frac{1}{2}(x_1 - x_0)(f_0 + f_1) \tag{7}$$

$$E_1(f) = -\frac{h^3}{12}f''(\xi), \quad \xi \in (x_0, x_1) \tag{8}$$

其中，$h = x_1 - x_0$，当 $n = 2$ 时，由公式(4)得到**辛普森公式**及其误差

$$I_2 = \frac{h}{3}(f_0 + 4f_1 + f_2) \tag{9}$$

$$E_2(f) = \frac{(x_2 - x_0)^5}{2880}f^{(4)}(\xi), \quad \xi \in (x_0, x_2) \tag{10}$$

其中 $h = \frac{1}{2}(x_2 - x_0)$。

当 $n = 4$ 时，由公式 (4) 得到常用的柯特斯公式及其误差

$$I_4 = \frac{2h}{45}(7f_0 + 32f_1 + 12f_2 + 32f_3 + 7f_4) \tag{11}$$

$$E_n(f) = -\frac{8}{945 \times 4^7}(x_4 - x_0)^7 f^6(\xi), \quad \xi \in (x_0, x_4) \tag{12}$$

其中 $h = (x_4 - x_0)/4$。

由公式 (8)，公式(10) 和 公式(12) 看出，当积分区间很小时，用 公式(7)，公式(9) 和 公式(11) 计算定积分可以取得很好效果。但是当积分区间很大时，误差较大，这时可采用复化求积公式。

复化梯形公式 将区间 $[a, b]$ 分为 n 个小区间，令 $h = \frac{b-a}{n}$，$x_i = a + ih$，$x_0 = a$，$x_n = b$，在每个小区间上使用梯形公式，并对 $i = 0, 1, \cdots, n-1$ 求和，就得到复化梯形公式及其误差

$$\int_a^b f(x)\,dx \approx T_n = \frac{h}{2}\left[f(a) + f(b) + 2\sum_{i=1}^{n-1} f(x_i)\right] \tag{13}$$

$$E_1^{(n)}(f) = -\frac{b-a}{12}h^2 f''(\xi), \quad \xi \in [a, b] \tag{14}$$

复化辛普森公式 将区间 $[a, b]$ 等分为 $2n$ 个小区间，令 $h = \frac{b-a}{2n}$，在每两个相邻小区间上使用辛普森公式，并对 $i = 0, 1, \cdots, n-1$ 求和就得复化辛普森公式及其误差：

$$\int_a^b f(x)\,dx \approx S_{2n} = \frac{h}{3}\left[f(a) + f(b) + 4\sum_{i=1}^{n} f(x_{2i-1}) + 2\sum_{i=1}^{n-1} f(x_{2i})\right] \tag{15}$$

$$E_2^{(n)}(f) = -\frac{h^4}{180}(b-a)f^{(4)}(\xi), \quad \xi \in [a, b] \tag{16}$$

高斯型公式 考虑 $\rho(x) \neq 1$ 的情形。公式 (2) 含有 $2(n+1)$ 个参数 (x_i 和 A_i, $i = 0, 1, \cdots, n$)，适当选择这些参数，可以使公式 (2) 的代数精度达到 $2n+1$。数值积分理论中一个基本定理断言：只要把结点 $x_0, x_1, \cdots, x_n$ 取为区间 $[a, b]$ 上关于权函数 $\rho(x)$ 的 $n+1$ 次正交多项式的零点，则内插型求积公式 (2) 可达到最高代数精度 $2n+1$，这时，相应的求积公式称为高斯型求积公式。

设 $Q_{n+1}(x)$ 是 $[a, b]$ 上关于权 $\rho(x)$ 的 $n+1$ 次正交多项式（即当 $i, j = 1, 2, \cdots$ 且 $i \neq j$ 时成立 $\int_a^b \rho(x)Q_i(x)Q_j(x)\,dx = 0$，则称 Q_i, Q_j 在 $[a, b]$ 上关于权 $\rho(x)$ 正交)，利用正交多项式的一些关系式和性质，可以导出 $x_0, x_1, \cdots, x_n$ 是 $Q_{n+1}(x)$ 的 $n+1$ 个零点，则高斯型求积公式的 $n+1$ 个系数为

$$A_i = \frac{1}{Q_{n+1}^{*\prime}(x_i)}\int_a^b \frac{\rho(x)Q_{n+1}^*(x)}{(x - x_i)}dx = \frac{a_{n+1}}{a_n Q_{n+1}^{*\prime}Q_n^*(x_i)}, \quad i = 0, 1, 2, \cdots, n \tag{17}$$

式中 $Q_{n+1}^*(x)$ 是 $[a, b]$ 上关于权函数 $\rho(x)$ 的$n+1$ 次规格化正交多项式，a_{n+1} 和 a_n 分别是 $n+1$ 次和 n 次规格化正交多项式 $Q_{n+1}^*(x)$ 和 $Q_n^*(x)$ 的首项系数。

高斯型积分公式 (2) 的截断误差为

$$E_n = \int_a^b \rho(x)f(x)\,dx - \sum_{i=0}^{n} A_i f(x_i) = \frac{f^{2n}(\xi)}{a_{n+1}^2(2n)!}\int_a^b \rho(x)Q_{n+1}^2(x)dx, \quad a < \xi < b \tag{18}$$

已经证明：当$f^{2n}(x)$ 在 $[a, b]$ 上连续，那么当 $n \to \infty$ 时，高斯型积分公式收敛于定积分，即

$$\lim_{n \to \infty} \sum_{i=0}^{n} A_i f(x_i) = \int_a^b \rho(x)f(x)\,dx$$

牛顿–柯特斯公式却不具备这种收敛性质。

上面对于一般权函数讨论了高斯型积分公式。不同的权函数有不同的具体高斯型积分公式。表 1 给出几种常用的高斯型求积公式及截断误差。用这些公式积分时，可用已给出的求积结点与系数表。

表 1　几种常用的高斯型求积分式

$[a,b]$	$\rho(x)$	$x_0,x_1,\cdots,x_n$	A_i	E_n
$[-1,1]$	1	勒让德多项式 $P_{n+1}(x)$ 的零点	$-\dfrac{2}{(n+2)P_{n+2}(x_i)P'_{n+1}(x)}$	$\dfrac{2^{n+1}[(n+1)!]^4}{(2n+3)[(2n+2)!]^3}f^{(2n+2)}(\xi)$
$[0,\infty)$	e^{-x}	拉盖尔多项式 $L_{n+1}(x)$ 的零点	$\dfrac{[(n+1)!]^2}{L'_{n+1}(x_i)L_{n+2}(x_i)}$	$\dfrac{[(n+1)!]^2}{(2n+2)!}f^{(2n+2)}(\xi)$
$[-1,1]$	$\dfrac{1}{\sqrt{1-x^2}}$	切比雪夫多项式 $T_{n+1}(x)$ 的零点	$\dfrac{\pi}{n+1}$	$\dfrac{\pi}{2^{2n+1}(2n+2)!}f^{(2n+2)}(\xi)$
$(-\infty,\infty)$	e^{-x^2}	埃尔米特多项式 $H_{n+1}(x)$ 的零点	$-\dfrac{2^{n+2}\sqrt{\pi}(n+1)!}{H'_{n+1}(x_i)H_{n+2}(x_i)}$	$\dfrac{\sqrt{\pi}(n+1)!}{2^{n+1}(2n+2)!}f^{(2n+2)}(\xi)$

例　用勒让德求积公式计算积分 $I(f)=\int_0^{\pi}e^x\cos(x)\,dx$。

$I(f)$ 的积分精确值为 $-12.070\,346\,316\,4$，用勒让德求积公式计算结果为

n	$I_n(f)$	E_n
2	−12. 336 210 465 70	2.66×10^{-1}
3	−12. 127 420 450 17	5.71×10^{-2}
4	−12. 070 189 490 29	-1.57×10^{-4}
5	−12. 070 328 535 89	-1.78×10^{-5}
6	−12. 070 346 331 10	1.47×10^{-8}
7	−12. 070 346 317 53	1.14×10^{-9}
8	−12. 070 346 316 39	$<5.0\times10^{-12}$

除上述方法外，还可采用龙贝格求积公式、三次样条求积公式、自适应求积公式等。奇异积分问题一般不能用前面所述方法计算，必须针对具体积分选择适当方法。

参考文献

1. 冯康，等. 数值计算方法. 北京：国防工业出版社，1978

2. 关治，陈景良. 数值计算方法. 北京：清华大学出版社，1990

（李晓梅）

shuzhi jisuan

数值计算（numerical computation）　有效使用数字计算机求数学问题近似解的方法与过程，或这些方法与过程连同有关理论所构成的学科。

数值计算是一门实用性很强的学科，近年来随着计算机的发展和广泛应用，许多计算领域的问题，如计算力学、计算物理、计算化学、计算经济学等新分支都可归结为数值计算问题。

简史　早在公元前14世纪的商代就基本形成十进制的记数法，春秋战国时代筹算已得到普遍应用，并形成和发展了算筹作为通用有效的计算工具，以后改进演化为算盘，相应发展了珠算方法。

数值计算在古代已取得了重要成果。早在公元1世纪，汉代的《九章算术》记载了开平方、开立方、解一元二次方程和三元一次方程的方法。公元5世纪南北朝时期，著名数学家祖冲之算得圆周率π为3. 141 592 6和3. 141 592 7之间，精确到小数点后7位，这项纪录保持了近千年。南宋数学家秦九韶于1247年写成《数书九章》，提出的联立一次同余式和高次方程数值解的秦九韶法，比西方Euler和Horner于1819年提出的算法早五百多年。

随着近代数学的形成和发展，数值计算也取得了相应的进步。20世纪40年代中后期，电子计算机的出现与迅速发展，有力推动了数值计算的研究与发展，解决了许多难度与规模都很大的计算问题，提出了许多新的数值方法，数值计算在整个科学技术和经济生活中的重要性得到前所未有的体现，并获得极大的发展，形成了一门新的学科分支。

数值计算与计算机的发展相辅相成，相互促进。大量数值计算的需要，促使计算机体系结构及性能不断更新，而计算机的发展又推动着数值计算方法的发展，每当计算机发生一次变革，数值计算也产生一次飞跃。为适应现代计算机的飞速发展，对数值计算提出了新的要求，对原有计算方法提出了重新评价、筛选、改造和创新。与此同时，也涌现了许多新概念、新方法，从而构成了现代数值计算的新含义，如计算机系统界面的多媒体化，给计算过程可视化提供了软件环境。

研究内容　数值计算研究有效使用计算机数值求解各种数学问题，包括离散型方程的数值求解和

连续系统离散化的数值求解,在数值求解数学问题时,需要考虑误差、收敛性和稳定性等问题。所谓关于给定计算问题的一个近似算法是收敛的,是指由该算法能产生近似解的一个无穷集合,这个集合按某种选定的距离能逼近精确解到任意程度。即对任给的 $\varepsilon > 0$,都能从该集合中找到与精确解的距离小于 ε 的近似解。误差是指连续系统离散化产生的方法误差(截断误差)和数值计算过程中产生的误差(舍入误差)。稳定性是指在执行数值算法的过程中,舍入误差的积累不影响产生可靠结果。此外,还要研究算法的计算复杂性(计算量大小为时间复杂性,存储量大小为空间复杂性)以及在使用计算机时,算法的自适应性。因此,误差、收敛性、稳定性、计算复杂性和自适应性是数值计算的基本问题,刻画了数值计算方法的可靠性、准确性、效率以及使用方便性,是数值计算必须研究的基本理论。

数值计算的研究内容随着计算机发展和应用范围的扩大而不断扩大,从数学类型看,它包括**数值逼近**、**数值微分与数值积分**、快速傅里叶变换、数值代数、**最优化方法**、**常微分方程数值解法**、积分方程数值解法、**偏微分方程数值解法**、**计算几何**、计算概率统计、并行算法等。

数值逼近是研究函数的离散逼近,用适合于计算机上计算的简单函数(如多项式,连分式等)逼近复杂函数。主要包括插值法、最佳一致逼近、最佳平方逼近与最小二乘逼近。各种代数插值是连续系统离散化的基础,理论与实践表明,分段低次插值比高次插值具有更好的稳定性。由于样条函数具有良好的特性,已成为函数逼近的重要工具而被广泛应用,样条逼近是函数逼近的一个重要研究方向。数值逼近也是生成初等函数子程序的重要方法。

数值微分与数值积分是在数值逼近基础上发展起来的计算微分与积分的数值方法。数值微分由于稳定性问题计算效果不理想,特别是对高阶导数和多元函数偏导数的计算效果更差。20 世纪 80 年代提出的新方法是用公式的数值求导或用符号运算的解析求导。数值积分常用的方法是数值稳定的,各种方法的差异在于计算量大小和自适应性,反常积分与高振荡积分要采用特殊的数值积分法。

快速傅里叶变换(FFT)是用三角函数逼近周期函数的递推方法,它把 N 个点变换的计算量从 $O(N^2)$ 下降为 $O(N\log_2 N)$,当 N 很大,效果尤为明显。FFT 算法为调和分析方法在谱分析、信号处理和图像处理等领域应用计算机开辟了道路,也开创了各种快速算法的先河。

数值代数是对有限维离散型方程的数值求解,内容包括**线性代数方程组数值解法**、**矩阵特征值问题数值解法**、一元方程求根、**非线性代数方程组数值解法**等。各种连续系统的离散化,最终归结为有限维离散型方程的数值求解。这类方程的数值解法分为两类:一类是迭代法,主要是构造收敛快、计算复杂性好的算法。对线性方程,人们针对不同类型方程特点,构造了行之有效的各种迭代法。对非线性方程,以牛顿法为代表的各种牛顿型迭代法已得到广泛应用,但由于这类方法只具有局部收敛性,因此,近 30 年来,具有大范围收敛的同伦连续法、单纯同伦算法、区间迭代法和单调迭代法等成为该领域的重要研究课题。另一类是直接法,它只对线性方程和极特殊的非线性方程有效,直接法由于舍入误差影响,方法的稳定性和方程是否病态等问题受到极大关注。当前病态方程组及大型稀疏方程组数值解法仍然是重要的研究方向。特征值计算包括标准特征值、广义特征值和反特征值问题的计算,反特征值问题由于条件的不同,提法各异,目前发展迅速并有广泛应用。

最优化方法是数值计算的一个迅速发展的领域,它研究在结构比较复杂的集合上泛函的极值问题。首先是规划问题,包括线性规划、非线性规划和动态规划等问题,经济学中许多问题可归结为这类问题。线性规划是理论成熟、应用最广泛的部分,基本解法是单纯形法。1984 年,N. Karmarker 提出了一种多项式时间的新方法,引起广泛注意。该方法的计算复杂度为 $O(n^{3.5}L^2)$,其中 n 为变量数,L 为输入长度。此外,运筹学和对策论中提出的极小极大问题及相应的计算方法也是最优化计算的研究内容。在经济和工程中求解大型最优化问题,目前已有可供使用的通用和专用的计算软件。

常微分方程数值解法包括初值问题与边值问题两类,初值问题的基本算法是龙格-库塔法和阿当姆斯法。近 30 年来,刚性问题的数值解法及其稳定性理论的研究有很大进展。边值问题的数值解法主要有两种,一种是转化为初值问题的打靶法;另一种是直接离散化的差分方法和变分方法。积分方程数值解法是直接将数值积分公式转化为离散化方程求解。

偏微分方程数值解法主要研究方程的离散化和对离散方程的求解,以及有关适定性、收敛性、稳定性、误差和自适应性等理论问题,使用计算机进行科学与工程计算往往涉及到偏微分方程的数值解法,

诸如天气预报、反应堆设计的核扩散研究、液体流动、超音速流及弹性力学、地球物理、地震勘探等。

偏微分方程数值解法有正演问题和反演问题两类。正演问题是给定方程和定解条件求偏微分方程的解,它主要是二维和三维的定常问题和非定常问题。微分方程的离散化主要有差分法和有限元法两大类。差分法以差商逼近导数,将微分方程离散为差分方程,这种方法简单,适用于任何类型的方程,同时便于机器实现。有限元方法基于等价的变分原理的形式,采取任意格网分片逼近的手段,把传统对立的差分方法与里茨-伽辽金变分方法有机地结合起来,发展成为解定常问题的主导方法,并推广到非定常问题。有限元法具有几何上灵活适应的突出优点,特别适合于解决复杂性大的问题。目前已有比较成熟的通用和专用有限元法计算软件,这些软件在众多科技领域和工程设计中已得到普遍应用。反演问题是在给定方程的模式下,已知其解或解的某一部分,要反推求该方程的系数和边界的形式等,起着导果求因的作用。这类问题在医疗卫生、无损探测、遥感遥测、地震勘探、地球物理等领域大量存在,它们通常是不适定的、病态的,有其特殊的难点。反演问题数值解法是一个正在开拓的新领域。

计算几何是数值计算的新分支,它研究静态或动态的几何形体和视象的离散化,以及逼近与生成的计算方法,是计算机辅助几何设计(CAGD)这门技术的数学基础。计算几何在计算机制图、计算机辅助设计、计算机辅助制造、计算机动画、计算机视象、机器人技术等众多领域有重要的应用。

计算概率统计主要研究随机数据的统计分析计算和概率系统模型的随机模拟计算。前者包含多元统计分析计算、时间序列分析计算、马尔可夫链计算和数值滤波等。后者最常用的是蒙特卡罗法。这方面的计算是根据实际问题的概率统计模型,对试验观测数据或随机模拟数据进行统计分析处理,给出实际问题性质的统计描述、统计控制或统计预测的数值结果。目前已研制出了众多的计算概率统计软件供用户使用。

数值计算软件是数值计算研究的另一个重要方面,每个数值软件实现一个特定的计算方法的标准程序模块,由一个或几个标准子程序组成。由于数值计算软件经常为用户使用,因此,它的研制除了要吸取一般软件理论和技术外,还有它自身的特点。首先要对计算方法进行认真选择,考虑方法的适用性、准确性、稳定性、计算量和存储量等因素。其次每个数值计算软件必须经过全面严格测试,以确保其正确性和可靠性。最后每个数值计算软件需有详尽的说明书,对软件的使用、适用范围及正确性给予确切说明。目前,已问世的数值计算软件涉及到数值计算各领域。随着数值计算方法的进一步发展和计算机的更广泛应用,数值软件包含的内容也将越来越广泛。

并行算法是数值计算的一个特别活跃的新领域,它随着并行计算机的发展而迅速发展。并行计算机包括单指令流多数据流计算机(SIMD)和多指令流多数据流计算机(MIMD)两类,与其相应的算法分为同步并行算法和异步并行算法两类。20世纪60年代末到70年代初,由于解偏微分方程及大型线性方程组等科学计算的需要,人们开始设计阵列式与向量式的新型结构计算机,开始研究同步并行算法。

1972年,阵列式计算机ILIACN-IV在美国宇航局投入运行,TI-ASC在欧洲运转,此后向量计算机CDCSTAR-100和CRAY-1等相继于1974年和1976年在美国投入运转。从1972—1981年这一时期,并行算法进入实践的阶段与具体并行机的相关性增强,并行算法复杂性分析与评估有了新的内容,并行数值计算软件开始研制,但集中在串行程序的向量化研究上。

1982—1991年是并行算法研究的兴盛期,在这一阶段,MIMD超级计算机系统CRAY X-MP,HEP与CRAY-2等相继投入运行,各类MIMD计算机系统如雨后春笋,特别是大规模并行SIMD计算机CM-2及MIMD计算机CM-5相继于1987年和1991年问世,一些大型科学与工程问题,如来自航天、化学、天体物理、材料科学、数据检索、石油勘探、天气预报等广泛领域中的问题,都能应用这种大规模并行计算机来求解。这时期并行计算的特点是:同步并行算法日益成熟并逐步完善;SIMD计算机语言开始形成以Fortran 90为基础的标准框架;异步并行计算成了热门研究课题。

1992年以后,并行计算开始转入标准化时期,这时期出现了规模更大速度更快的并行机,1994年思维公司推出了它的第五代并行处理系统CM-5E,它的峰值运算速度达到每秒830亿次浮点运算,CRAY研究公司也于1994年宣布了世界上最先进的并行处理机CRAY T3D系统,该系统由2048个处理机组成,其峰值运算速度达每秒3072亿浮点运算,这些对并行算法引起了深刻的变革,出现了一些

新的研究方向,如格子气方法、基于神经网络的并行算法以及演化算法等。而并行计算语言的标准化和并行虚拟机的研究为并行算法的研究开辟了道路。

(李庆扬　康立山)

shuzhi jisuan wucha fenxi

数值计算误差分析(error analysis of numerical computation)　关于数值计算中误差的产生与传播以及如何分析与控制各种误差的方法与过程。

数据近似值与精确值之差是衡量数据可靠性和精确度的重要方面。应用数值方法在计算机上求解实际问题时,由于模型、测量手段和计算工具等方面的限制,以及计算方法的差异,所得结果往往不是所考虑对象的准确值,而是近似值。

设近似值为 a,准确值为 x,则 $e = x - a$ 称为近似值 a 的误差。通常不可能得到准确值 x,也就不能算出误差 e 的值,而只能根据数据计算出误差 e 的绝对值的上界 ε, 即

$$|e| = |x - a| \leqslant \varepsilon$$

称 ε 为近似值 a 的**绝对误差**限或误差限,而准确值 x 的范围为

$$a - \varepsilon \leqslant x \leqslant a + \varepsilon$$

工程技术上常将上式表示为 $x = a \pm \varepsilon$。

在许多情况下误差限的大小还不能完全刻画一个近似值的准确程度,它还与准确值 x 的大小有关,比值 e/x 称为近似值 a 的**相对误差**,并用记号 e_r 表示,即

$$e_r = \frac{e}{x} = \frac{x - a}{x}$$

实际计算中常取 $e_r \approx \frac{x - a}{a}$, 并把相对误差 e_r 的上界称为相对误差限,记作 ε_r, 即 $|e_r| \leqslant \varepsilon_r$, 计算时取 $\varepsilon_r = \frac{\varepsilon}{|a|}$。

数值计算的误差分析旨在估计数据计算结果的误差限。误差按其来源可分为模型误差、观测误差、截断误差和舍入误差等。

模型误差　数值计算首先要建立数学模型,它是对被描述的实际问题进行抽象、简化得到的,因而是近似的,由此产生的误差称为模型误差。

观测误差　数学模型中常包含某些参量,比如长度、温度、密度等,需要通过观测确定它们,由此产生的误差称为观测误差。

截断误差　数学模型中的表达式往往很复杂,为了在计算机上求解,必须提供合适的数值计算方法,这种方法是通过对原数学模型的表达式作某种近似而产生的,其误差称为截断误差或方法误差。如用 $x - \frac{x^3}{3!}$ 近似 $\sin x$ 时,截断误差为

$$\sin x - \left(x - \frac{x^3}{3!}\right) = \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots$$

舍入误差　用计算机进行计算时,由于机器字长有限,原始数据及计算过程中产生的数据,其位数超过规定位数时都要进行舍入,这样产生的误差称为舍入误差。

在数值计算中与误差有密切联系的另一个概念是有效数字。

有效数字　若一个数 x 的误差不大于该数某位数字的半个单位,则从非零数字最左边第一个数字起,到这一位数字止都是该数的有效数字,其个数称为该数有效数字的位数。例如数

$$x = \pi = 3.141\,592\,6\cdots,$$

若取近似值 $a = 3.14$,它有三位有效数字, $|x - 3.14| \leqslant \frac{1}{2} \times 10^{-2}$;若取 $a = 3.1416$, 它有5位有效数字, $|x - 3.1416| \leqslant \frac{1}{2} \times 10^{-4}$。一般地,设 x 的近似值 $a = \pm 10^m \times 0.a_1 \cdots a_n$,其中 $a_i (i = 1,2,\cdots,n)$ 是0~9的一个数字, $a_1 \neq 0$,m 为整数,且 $|x - a| \leqslant \frac{1}{2} \times 10^{m-n}$, 则近似值 a 相对数 x 具有 n 位有效数字。实际上,一个数按四舍五入规则得到的规格化近似数,其每一位数都是有效数字。显然,有效数字位数与小数点位置无关,有效位数越多绝对误差与相对误差就越小。

按有效数字概念,以下数字是有区别的,79.06,79.0600,前者为4位有效数字,后者为6位有效数字。

数值计算的误差分析是针对已建立数学模型的数值计算方法进行的,它是一个重要而复杂的问题,如果方法是近似的,就需要进行截断误差分析,不同方法有不同的截断误差估计公式。由于原始数据有误差,而每一步运算又会产生新的舍入误差,并传播前面各步已引入的误差,所以逐步分析误差是可行的。误差分析是估计整个计算过程积累误差的界,以判断数值计算结果的可靠性,这是一个很复杂的问题,目前尚无统一的方法,常用的误差分析方法有以下几种:

向前误差分析法与向后误差分析法　假设所讨论的方法由某个公式表达,某个结果 x 由已知量(原始数据或先前已算出的量) $a_1,a_2,\cdots,a_n$ 经过基本算术运算来确定,可写成

$$x = g(a_1,a_2,\cdots,a_n)$$

由于计算中产生舍入误差,实际计算出的值 a 与它的准确值 x 不同,向前误差分析是根据误差运算规则对每一步运算找出舍入误差界,随计算过程逐步向前分析,直至估出最后舍入误差 $|x-a|$ 的上界,这种误差分析方法只能应用于十分简单的情形。

向后误差分析则是把舍入误差与导出 a 的已知量 $a_1,a_2,\cdots,a_n$ 的某种摄动(微小误差)联系起来,即对某个 a_i 引进摄动量 ε_i,使得由浮点运算得到等式

$$a = g(a_1+\varepsilon_1,\cdots,a_n+\varepsilon_n)$$

并推出这些 ε_i 的界(ε_i 不是唯一的,且无须求出 ε_i 的具体值),然后利用摄动理论估计最后舍入误差 $|x-a|$ 的界。向后误差分析是一种先验性估计,20 世纪 60 年代初,J. H. Wilkinson 在研究矩阵计算的误差分析时有较系统的讨论,并取得了较大进展,奠定了向后误差分析的基础。它是计算机上各种数值计算最常用的误差分析手段,J. H. Wilkinson 在其著作《代数过程的舍入误差》和《代数特征值问题》中,对数值计算中浮点运算的误差界及矩阵运算和数值代数求解舍入误差分析有详细的讨论。

区间分析法　20 世纪 60 年代中期提出了一种利用区间算法进行误差分析的方法,它把参与运算的数 $a,b,c,\cdots$ 看成为分属于某些区间量 $A,B,C,\cdots$,通过对区间量运算求得结果 x 所属区间 X,即求得近似值与误差。例如,设 a^*,b^* 为准确值,a,b 是相应近似值,由于 $|a^*-a|\leqslant\delta a$, $|b^*-b|\leqslant\delta b$,则

$$a^* \in A = [a-\delta a, a+\delta a]$$
$$b^* \in B = [b-\delta b, b+\delta b]$$

利用 a^*,b^* 所在区间 A,B,按区间运算规则可得到 a^*,b^* 运算结果 x 的区间 $X=[\underline{x},\bar{x}]=[c-\delta c, c+\delta c]$,则 c 为 x 的近似值,δc 为误差限,即 $|x-c|\leqslant\delta c$。这就是区间分析法的基本思想。

近 30 年来,由区间运算发展起来的区间数学已形成一个新学科,用区间方法求解数值计算问题,得到的结果也是一个区间量,它本身就包含了所求结果的近似值及其误差限。

概率分析法　通常对近似数运算结果误差限的估计总大于实际的误差,实际上误差分布有随机性,不会经常地达到上界。因此,利用概率统计的方法,将数据和运算中的误差视为适合某种分布的随机变量,并由此推出计算结果的误差分布,常常可以使误差估计更接近实际,这种误差分布称为误差的概率界。

数值计算的误差分析目的是保证方法产生符合精度要求的可靠结果,但对大量方法要定量分析舍入误差积累是非常复杂困难的,上面提供的方法往往也是很难实现的。因此,对舍入误差是否影响计算结果的可靠性进行定性分析是非常重要的,这就是方法的数值稳定性问题。一个方法如果在执行它的过程中,舍入误差的积累不影响产生可靠的结果,则称该方法是数值稳定的,否则称为数值不稳定的。不稳定方法是不能使用的,判断方法数值稳定的一个准则是原始数据的微小变化只会引起最后结果的微小变化。实用上只要方法是数值稳定的,就不必再对它的舍入误差进行定量分析。

参考文献

1. 关治,陈景良. 数值计算方法. 北京:清华大学出版社,1990

2. Wilkinson J H. Rounding error in algebraic processes. Prentice-Hall, 1963

3. Wilkinson J H. 代数特征值问题. 石钟慈,邓建新,译. 北京:科学出版社,1987

(李庆扬　周树荃)

shuzhi weifen

数值微分(numerical differentiation)　关于如何根据函数在一些离散点的值来推算它在某点的导数的近似值的方法。通常用差商来代替微商,或用某个能近似代替该函数的较简单的函数(如多项式函数、样条函数)的相应导数作为所求导数的近似值。在微分方程中常用数值微分将方程离散化以求数值解。

通常采用多项式插值来求数值微分。设函数 f 在 $n+1$ 个等距点 $x_v=a+vh(v=0,1,\cdots,n)$ 上的值 $f_v=f(x_v)$ 为已知,通过低次插值可导出一些最基本和最常用的数值微分公式。例如,两点公式 $f'(x_i)=\dfrac{f_{i+1}-f_i}{h}+O(h)=\dfrac{f_i-f_{i-1}}{h}+O(h)=\dfrac{f_{i+1}-f_{i-1}}{2h}+O(h^2)$,三点公式 $f''(x_i)=\dfrac{f_{i+1}-2f_i+f_{i-1}}{h^2}+O(h^2)$

等等。此外,根据具有 $n+1$ 个等距结点的拉格朗日插值公式,还可导出在结点 $x_i(i=0,1,\cdots,n)$ 上的较为一般的数值微分公式

$$f'(x_i) \doteq \frac{1}{h}\sum_{v=0}^{n} A_{i,v}f_v,\quad f''(x_i) \doteq \frac{1}{h^2}\sum_{v=0}^{n} B_{i,v}f_v$$

其中 $A_{i,v}$,$B_{i,v}$ 仅与 n,i,v 有关,相应的截断误差分别为

$$R'(x_i) = C_i h^n f^{(n+1)}(\xi_i)$$

$$R''(x_i) = \begin{cases} 2C_i\left(\sum_{v=1}^{i}\frac{1}{v} - \sum_{v=1}^{n-i}\frac{1}{v}\right)h^{n-1}f^{(n+1)}(\eta_i) \\ \left(i \neq \frac{n}{2}\right) \\ 2C_i h^n f^{(n+2)}(\eta_i)/(n+2) \qquad \left(i = \frac{n}{2}\right) \end{cases}$$

其中 $C_i = (-1)^i i!(n-i)!/(n+1)!$;$\xi_i,\eta_i \in (a,a+nh)$。因此,当结点的个数 $n+1$ 固定时,间距 h 越小,则截断误差也越小。但是,系数绝对值之和 $\frac{1}{h}\sum_{v=0}^{n}\left|A_{i,v}\right|$ 与 $\frac{1}{h^2}\sum_{v=0}^{n}\left|B_{i,v}\right|$ 随 h 的变小而剧增,所以函数值 f_v 的舍入误差对近似导数的影响也随 h 的变小而剧增。因此,h 并非越小越好,而是要适中,这是数值微分不同于某些插值之处。如果函数 f 有很好的可微性,即存在绝对值不太大的较高阶导数,则宁取间距稍大而个数稍多的结点。当 f 在结点分布的整个区间上的可微性不太好时,采用样条插值来进行数值微分比采用多项式插值更适宜,只是计算量要大得多。

如果数据 f_v 具有不容忽视的随机误差,且对应的自变量分布甚密,那么就应该用曲线拟合来代替上述函数插值,然后用拟合曲线的导数作为函数 f 的导数的近似值。这样求得的导数叫做磨光的导数。

参考文献

冯康,等. 数值计算方法. 北京:国防工业出版社,1978

(沈毅)

shuzhi

数制(number system) 用一组统一的符号和规则来表示数的方法。它涉及记数制,定点数和浮点数的表示,数的原码、补码、反码表示及其相应的运算规则。

记数制

用若干个数码的组合来表示 1 个数,每个数码的值与其在数中所处的位置有关。例如一位十进制数可以用 0,1,2,…,9 的 10 个数码中的任一个来表示,一位二进制数可以用 0 或 1 来表示。每个数码处在不同的位置时所代表的数值不同,每个数码所表示的数值等于该数码本身乘以 1 个与它所在位置有关的常数,这个常数称为权。一个数码可以选用的值的个数就是相应的记数制的基数,十进制数的基数为 10,二进制数的基数为 2。

十进制数的值可用 1 组有序的数码或多项式来表示,例如:

$$586.13 = 5\times10^2 + 8\times10^1 + 6\times10^0 + 1\times10^{-1} + 3\times10^{-2}$$

式中每个数位的权分别为 $10^2,10^1,10^0,10^{-1}$ 和 10^{-2}。

其他进位制的表示方法雷同。

在计算机中,如果要求基本电子元件具有 10 个稳定状态来分别表示数码 0~9 是很困难的,但要求它们具有两个稳定状态来分别表示 0 和 1 却是容易实现的,而且二进制数的运算规则比十进制简单,因此在计算机中广泛采用二进制数。

二进制数可用一组有序数码或多项式表示,举例如下:

$$1011.01 = 1\times2^3 + 0\times2^2 + 1\times2^1 + 1\times2^0 + 0\times2^{-1} + 1\times2^{-2}$$

式中每个数码的权分别为 $2^3,2^2,2^1,2^0,2^{-1}$ 和 2^{-2}。

在计算机中的数有时也用八进制或十六进制表示,通常用 3 位二进制码来表示 1 位八进制数码,用 4 位二进制码来表示 1 位十六进制数码。

十进制、二进制、八进制和十六进制数如表 1 所示。表中 A,B,…,F 分别表示十六进制数的 10,11,…,15。

表 1 各种进位制数

十进制数	二进制数	八进制数	十六进制数
0	0000	00	0
1	0001	01	1
2	0010	02	2
3	0011	03	3
4	0100	04	4
5	0101	05	5
6	0110	06	6
7	0111	07	7
8	1000	10	8
9	1001	11	9
10	1010	12	A

续表

十进制数	二进制数	八进制数	十六进制数
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

通常输入计算机的是十进制数，用4位二进制数来表示1位十进制数，称为**BCD码**或二-十进制码（二进制编码的十进制数），BCD码是有权码，从高位开始各位的权分别为8，4，2，1，所以又叫8421码。对于一位十进制数还有**余3码**和**格雷码**等表示方式，如表2所示。余3码和格雷码是无权码，余3码由8421码加0011得来。格雷码有多种形式，它的编码规则是使相邻两数码之间只有一位不同，表2中列出两种格雷码。也可用格雷码构成四进制或八进制计数器等。由格雷码表示的数据不适用于算术运算。

表2　十进制数编码

十进制数码	8421码	余3码	格雷码	
0	0000	0011	0000	0000
1	0001	0100	0001	0100
2	0010	0101	0011	0110
3	0011	0110	0010	0010
4	0100	0111	0110	1010
5	0101	1000	1110	1011
6	0110	1001	1010	0011
7	0111	1010	1011	0001
8	1000	1011	1001	1001
9	1001	1100	1000	1000

二-十进制数输入计算机后，通常转换成二进制数，然后按二进制运算规则进行运算，在结果输出前转换成十进制数，然后再输出。某些计算机设置有十进制运算指令，可直接对十进制数进行运算。

定点数表示

在处理机的运算部件中，**定点数**的小数点位置是固定的，而且是隐含的（或约定的）。通常将小数点位置固定在数值的最右端或最左端，前者称为定点整数，后者称为定点小数。如果最左位定义为符号位，则定点整数的小数点位于数值部分之后，定点小数的小数点位于符号位之后，数值部分之前。例如01001，当默认为定点整数时，其值为+1001；当默认为定点小数时，其值为+0.1001。符号位为0时表示正数，为1时表示负数，-1001的机内码可表示为11001。

在计算机中，有时也可将数定义为不带符号的数，此时最左位是数据的最高位。

由于用户的初始数据、中间结果或最后结果可能在很大范围内变动，如果机器用定点整数或定点小数来表示数，程序员不得不在运算的各个阶段预先引入比例因子，将数放大和缩小，这会带给程序员很多麻烦。如果比例因子的选择没有达到恰到好处的理想情况，则运算时数据不是容易溢出就是容易丢失精度，因此某些处理机使用了小数点位置可以浮动的二进制浮点数表示形式。

浮点数表示

用**浮点数**表示的数分为**阶码**（或称为指数）和**尾数**两部分。阶码用二进制定点整数表示，尾数用二进制定点小数表示，阶码的长度决定数的范围，尾数的长度决定数的精度。例如+110100的数值等于$2^6 \times 0.110100$，阶码为+110，尾数为+0.110100，其浮点数的表示形式为

0	0	1　1　0	1　1　0　1　0　0
尾数符号	阶码符号	阶码	尾数

大多数计算机的浮点数表示符合IEEE 754标准，这时需对上述表示形式进行修正（参见**浮点数标准**）。

当尾数的绝对值大于或等于1/2时，称为规格化浮点数。反之，就不是规格化数。考虑到尾数有正、负两种情况，如果尾数用补码表示，则可归纳如下：凡是尾数最高位与尾数符号位不同的数为规格化数，即尾数符号位为0，且尾数最高位为1；或尾数符号位为1，且尾数最高位为0的数为规格化数。在计算机内存储的浮点数一般为规格化数，在这种情况下，为了增加数据精度，可以将最高位规定为隐含位。计算机的阶码通常用补码或移码（增码）表示。

数的原码、补码和反码表示

以正负符号和绝对值来表示的数称为真值，计算机中的数称为机器数。在计算机中，分别用0和1来表示正号和负号。例如真值+0.1101，机器数也为0.1101；真值-0.1101，机器数为1.1101。在数的符号用0或1表示之后，根据运算需要，对数值

部分可以有不同的处理方法，从而产生了**原码**、**补码**和**反码**3种表示方式。

1. 原码　用最高位表示符号：0表示正数，1表示负数；有效数值部分用二进制绝对值表示，原码表示与上述机器数表示形式一致。

设真值x为定点小数，原码用$x_0x_1x_2\cdots x_n$表示，且x_0为符号位，则

$$[x]_{原}=\begin{cases}x, & 1-2^{-n}\geqslant x\geqslant 0\\ 1-x=1+|x|, & 0\geqslant x\geqslant -(1-2^{-n})\end{cases}$$

设真值x为定点整数，原码用$x_nx_{n-1}\cdots x_1x_0$表示，且x_n为符号位，则

$$[x]_{原}=\begin{cases}x, & 2^n-1\geqslant x\geqslant 0\\ 2^n-x=2^n+|x|, & 0\geqslant x\geqslant -(2^n-1)\end{cases}$$

零值有两种：正零(00…0)和负零(100…0)。

如果数用原码表示，对两个数进行加减运算时，需要判定它们的符号和绝对值大小以后才能确定操作关系和操作类型，例如两个正数相减，大数应作为被减数；两个不同符号的数相减，应执行加操作，而且结果的符号还要进行单独处理。为了简化加减法运算，可采用补码或反码来表示数。

2. 补码　补码的正数与原码相同，负数则不同。补码的定义如下：

设真值x为定点小数，补码用$x_0x_1x_2\cdots x_n$表示，且x_0为符号位，则

$$[x]_{补}=\begin{cases}x, & 1-2^{-n}\geqslant x\geqslant 0\\ 2+x=2-|x|, & 0>x\geqslant -1\\ & (\mathrm{mod}\ 2)\end{cases}$$

设真值x为定点整数，补码用$x_nx_{n-1}\cdots x_1x_0$表示，且x_n为符号位，则

$$[x]_{补}=\begin{cases}x, & 2^n-1\geqslant x\geqslant 0\\ 2^{n+1}+x=2^{n+1}-|x|, & 0>x\geqslant -2^n\\ & (\mathrm{mod}\ 2^{n+1})\end{cases}$$

零只有1种表示方式，即为正零(000…0)。

定点小数的补码运算具有以下特点：

$$[X+Y]_{补}=[X]_{补}+[Y]_{补}\qquad(\mathrm{mod}\ 2)$$

$$[X-Y]_{补}=[X]_{补}+[-Y]_{补}\qquad(\mathrm{mod}\ 2)$$

定点整数的补码运算应采用mod 2^{n+1}。

3. 反码　反码的正数与原码相同，负数则不同，反码的定义如下：

设真值x为定点小数，反码用$x_0x_1x_2\cdots x_n$表示，且x_0为符号位，则

$$[x]_{反}=\begin{cases}x, & 1-2^{-n}\geqslant x\geqslant 0\\ (2-2^{-n})+x, & 0\geqslant x\geqslant -(1-2^{-n})\\ & \mathrm{mod}\ (2-2^{-n})\end{cases}$$

设真值x为定点整数，反码用$x_nx_{n-1}\cdots x_1x_0$表示，且x_n为符号位，则

$$[x]_{反}=\begin{cases}x, & 2^n-1\geqslant x\geqslant 0\\ (2^{n+1}-1)+x, & 0\geqslant x\geqslant -(2^n-1)\\ & \mathrm{mod}\ (2^{n+1}-1)\end{cases}$$

零有两种表示：正零(000…0)和负零(111…1)。

从以上定义可得出：

(1) 负数的反码的符号位与原码相同，但数值部分的每一位都相反，例如已知x为负数，$[x]_{原}=1.0110$，则$[x]_{反}=1.1001$。

(2) 如果x为负数，比较$[x]_{反}$和$[x]_{补}$的公式，可得出：

当x为小数时，$[x]_{补}=[x]_{反}+2^{-n}$；当x为整数时，$[x]_{补}=[x]_{反}+1$。即将$[x]_{反}$的最低位加1，可得$[x]_{补}$。

浮点数的阶码常用**移码**(增码)来表示，移码的性质如下：

设阶码真值x为定点整数，其数值范围为$-2^n\sim+(2^n-1)$，移码用$x_nx_{n-1}\cdots x_1x_0$表示，则

$$[x]_{移}=2^n+x,\qquad 2^n>x\geqslant -2^n$$

将这一定义与定点整数补码的定义相比，可得出：

当$0\leqslant x\leqslant 2^n-1$时，$[x]_{移}=2^n+x=2^n+[x]_{补}$；当$-2^n\leqslant x<0$时，$[x]_{移}=2^n+x=(2^{n+1}+x)-2^n=[x]_{补}-2^n$。

这表明，取$[x]_{补}$的符号的相反值，保持其数值部分不变，即可得$[x]_{移}$。例如：

$x=+1011$　$[x]_{补}=01011$　$[x]_{移}=11011$；

$x=-1011$　$[x]_{补}=10101$　$[x]_{移}=00101$。

零只有1种表示：100…0。

当阶码$x=-2^n$时，$[x]_{移}=000\cdots0$，此为机器能表示的最小绝对值。一般当$x\leqslant-2^n$时，不管尾数值为多少，一律以机器零表示，此时阶码与尾数均以全0表示。　(王爱英)

shuzi bowuguan

数字博物馆(digital museum)　采用信息技术将文化遗产所涉及的各种信息以数字化形式进行采集、保存和管理，并通过**计算机网络**等通信手段为用户提供数字化的展示、教育和研究等多种服务的**应用软件系统**。

数字化展示的文化遗产称为数字博物馆的藏

品。数字博物馆的藏品概念比传统博物馆要丰富得多。一切文化遗产,有形的和无形的,物质的和非物质的,都可以成为数字博物馆的藏品。比如,民间收藏的文物,还有一些有历史价值的建筑物和壁画,它们可能属于某个实体博物馆,也可能不属于实体博物馆,但经数字化后都能成为数字博物馆的藏品;某些著名或濒临失传的音乐、舞蹈、木偶、地方戏曲、手工制作工艺等艺术和绝活,乃至现有的各类数字创作行为中具有一定意义和价值的数据资源也可以放到数字博物馆中。

随着**互联网**的发展,数字博物馆在 20 世纪 90 年代兴起。许多国家,特别是信息产业比较发达和重视文化的国家和地区,都启动了数字博物馆的有关计划或工程。美国国会图书馆 1990 年推出了“American Memory 计划”,对馆内文献、手稿、影音等藏品数字化。加拿大整合许多博物馆的藏品资源,开始建设加拿大虚拟博物馆,提供在线展览等多种服务。日本 1996 年实施了“次世代数字典藏系统研究与发展专案”,研究数字典藏的相关技术。许多著名的博物馆也纷纷开展了馆藏品的数字化工作。如美国纽约大都会博物馆、法国卢浮宫博物馆和凡尔赛宫博物馆、英国大英博物馆、俄罗斯国家历史博物馆、日本京都国立博物馆等。我国也十分重视数字博物馆建设,教育部 2001 年启动了大学数字博物馆工程,建设了十余所各具特色的大学数字博物馆。2006 年中国科协开始建设中国数字科技馆。中国国家博物馆、故宫博物院、台北故宫博物院等也开始了数字博物馆建设。

与实体博物馆相比,数字博物馆除了藏品的内涵有很大发展外,还有如下主要特点:①以数字形式表达和存储文化遗产的各方面信息,拓宽了信息的使用范围,避免了实体博物馆面临的藏品收集费用高、展出数量受建筑空间限制等问题。②藏品展示可以利用**多媒体技术**和**虚拟现实**技术在时间和空间维度上任意延伸,再现藏品所处时代人们生产和生活的场景。③通过不同领域数字藏品的整合,可以跨越学科知识的界限,实现跨领域的知识融合,打破了不同类型实体博物馆的藏品只能从某个角度展示的局限,拓宽了博物馆的内涵和外延。④摆脱了实体博物馆所必需的建筑、陈列、参观时间等条件的束缚,使任何人在任何时间、任何地点都能够获取特定的信息服务。⑤利用各种信息检索技术,可以为用户提供信息资源的个性化服务,让观众“看什么、有什么”,而不是实体博物馆的“展什么、看什么”,增强了用户的主动参与性。

数字博物馆涉及诸多学科,包括计算机科学技术、**软件工程**、文物与博物馆学、考古学等。围绕藏品信息的采集、组织、管理与服务,其主要技术有:①资源的数字化制作　包括藏品信息的多层面采集、转换、组织、压缩、整理等数字化技术以及数字内容的标注、分类、关联、浓缩、清洗等深加工技术。②海量多结构数据的存储与管理　包括数字对象的组织、数字对象仓储和仓储分布式体系等技术以及海量跨媒体数据的知识表达与索引、智能处理、融合与集成和跨媒体搜索等技术。③数字博物馆服务　包括交互式的数字藏品二维和三维展现技术以及基于主题词检索、元数据检索、基于内容的多媒体检索等多种检索方式和虚拟展览的定制服务、个性化服务和各种增值服务技术。④数字藏品的安全保护　主要有访问权限控制、数字水印、加密等技术。⑤技术规范　涵盖数字藏品资源的数字化加工、组织、管理、互操作、安全和服务等方面的标准。

未来的数字博物馆将通过历史场景的虚拟再现、互动参与及体验,让观众自主发现历史、感受文化传统。

参考文献

1. 徐世进,陈红京,董少春. 数字博物馆概论. 上海:科学技术出版社,2007

2. 齐越,沈旭昆. 博物馆数字资源的管理与展示. 上海:科学技术出版社,2008

（齐越）

shuzi cijilu

数字磁记录(digital magnetic recording)

以正向或反向饱和磁化形式在移动的磁性薄膜介质上按磁道存储数据信息的技术。磁记录因其所记录的信号不同分为两种:一种是模拟磁记录,被记录的信号是模拟信号,如通常的录音、录像和记录供监测的振荡信号;另一种是数字磁记录,被记录的是经过编码的脉冲信号。记录媒体上呈现出一串饱和的磁化状态翻转,形成一条磁道。在移动速度恒定时,翻转间距的长短与一定频率的数字信号的编码方式相对应。

数字磁记录技术是磁表面存储器(参见**磁存储器**)的基础。今天,它已发展成为由应用磁学、精密机械学、自动控制、半导体技术、计算机技术等多学科综合而成的一门在计算机领域中起重要作用的技术。

第一次出现磁记录是在 1898 年,丹麦人 V.

Poulsen 发明了用单极型磁头在钢丝上录音的录音机。1936 年,德国人 F. Pfleumer 用细粒磁粉涂布的磁带代替钢丝,使用环形磁头记录,改进了录音机,同时也开创了磁性材料与基底材料分离的先例,使磁性能与机械性能得以分别独立发展。

20 世纪 40 年代中期,计算机出现之后,使用磁鼓和磁带作存储器,数字磁记录设备成了计算机外存储器的主流,数字磁记录理论与技术获得了极大的发展,继而出现了速度快、记录密度高、存储容量大的**磁盘存储器**。同时,与之密切相关的磁头、磁记录媒体、读写通路、磁记录编码、纠错码以及磁头定位与跟踪技术等也得到了同步发展,在全世界形成了以磁盘存储器为主体的庞大的磁记录工业。

特点　数字磁记录有以下特点。

(1) 数据的写入与读出速度快,因而数据传输速率高。

(2) 磁化状态翻转密度高。从原理上看,记录密度可以做到很高,提高磁道密度和位密度还是今后追求的目标。

(3) 可方便地用交、直流抹除媒体上原有的信息;或在重写时覆盖,覆盖后原有信息的残余度很小。

(4) 能永久保留记录的信息,不致因掉电丢失。

(5) 读出信号为磁化媒体各层的集体贡献,不单纯取决于表层状态,故原始误码率较低,降低了数据检验纠错的难度。

(6) 原材料供应丰富,价格低廉。

上述特点决定了磁记录技术在信息存储技术中处于主导地位,但该技术在使用中对环境的要求较高,要求防尘、防振、防止强磁场干扰。

数字磁记录过程　磁记录包含写入过程、读出过程和抹除过程。

1. 写入过程　将待记录的二进制数经编码和电流放大后送入磁头线圈,在磁头铁心的前隙两侧产生磁压降,形成缝隙磁场。利用这个密集的磁场磁化具有永磁性质的磁层,使反映二进制数的写电流转换为可永久保留的磁化状态翻转图形,从而完成写入过程。由于线圈中流过的是极性按编码变换的电流,写入过程中媒体匀速移动,所以形成了由极性翻转的磁化单元所组成的磁迹,称之为磁道。磁道上 N 极与 S 极间的最小磁化状态翻转间距等于编码记录序列中两个 1 的相应间隔,如图 1 所示。

2. 读出过程　媒体上的磁化图形还原为二进制数的过程称作读出过程,图 2 为以环形磁头读出时读出过程的示意图,当有剩磁的磁化图形通过磁头下方时,磁头线圈通过铁心与剩磁交链,变化的磁通在线圈中感应出电压信号 $e(t)$。在磁化翻转中心处,读出电压出现正、负峰值,两次翻转中间位置的信号幅度为零。若以彼此极性反向排列的永久磁铁模拟磁化翻转图形,用磁头在相对匀速运动中读取其剩磁,则从示波器上显示的读出波形如图 3 所示。此波形不是正弦波,而是与磁头磁场有关的特殊波形。它有如下性质。

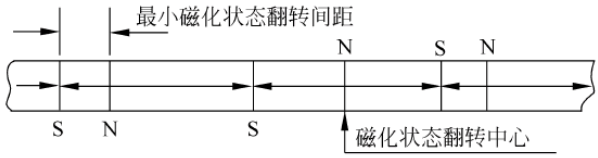


图 1　纵向磁记录磁化翻转图形

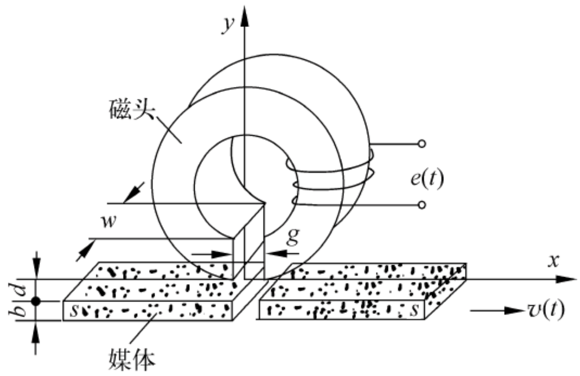


图 2　读出过程

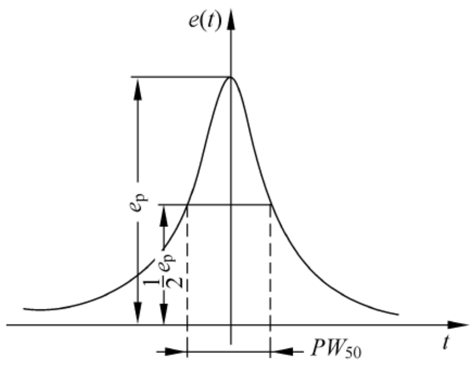


图 3　读出信号波形

(1) 读出信号幅度 e_p 与媒体移动速度 $v(t)$ 成正比。当 $v(t)$ 是常数,即速度恒定时,可写成 $e_p = kv$,其中 k 为比例系数。

(2) 图 3 中曲线 $e(t)$ 与 t 轴间所包含的面积是

常数。即不管 e_p 幅度如何变化，$\int_{-\infty}^{+\infty} e(t)\,dt$ 是一个常数。由此可见，当 e_p 大时，波形是瘦高形的，而当 e_p 小时，波形是矮胖形的。通常用 $(1/2)e_p$ 处的波形宽度（称为 PW_{50}）来表示波形形状特征。

（3）按照法拉第定律，有

$$e(t) = -N\frac{d\varphi}{dt} = -N\frac{d\varphi}{dx}\frac{dx}{dt} = -Nv\frac{d\varphi}{dx}$$

式中，N 为线圈匝数；$\frac{d\varphi}{dx}$ 为媒体上剩余磁通沿 x 方向的变化率，它与媒体上的磁化状态有关。对于一根磁棒，$\frac{d\varphi}{dx}$ 最大值出现在它的顶部（相当于磁记录中磁化状态的翻转中心），$\frac{d\varphi}{dx}=0$ 出现在它的中点（当于磁记录中磁化单元的中间位置）。也就是说，$e(t)$ 的最大值 e_p 出现在磁化状态翻转中心通过磁头的时候。

（4）信号幅度与磁头前极面至媒体表面间的距离 d 有关。距离缩小则信号幅度增加，反之则衰减。对于软基底的记录媒体，如磁带、软磁盘、磁卡片等，多采用接触式记录，d 几乎等于零。对于硬基底的记录媒体，如硬磁盘、磁鼓等，多采用非接触式记录，但 d 很小，如硬磁盘驱动器中，现在已达到 5 nm 以下。

（5）信号幅度与磁头在媒体上覆盖的宽度成正比例。

（6）磁化状态翻转密度与读出信号波形的半幅宽度 PW_{50} 有关。显然 PW_{50} 小，信号波形窄，磁化状态翻转密度就高。从理论分析推导可以得到

$$PW_{50} = \sqrt{(d+a)^2 + (g/2)^2}$$

式中，d 为磁头与媒体间的距离；g 为磁头前隙宽度；a 为磁化翻转过渡区参数，它与媒体厚度 b 有关。半幅宽度可用以估测媒体和磁记录系统的性能，它与磁化状态翻转密度有如下的近似关系

$$\frac{1}{PW_{50}} \approx C_{-3dB}$$

或

$$\frac{1.38}{PW_{50}} \approx C_{-6dB}$$

C_{-3dB} 和 C_{-6dB} 分别指读出幅度允许衰减 3 dB 和6 dB 时的记录密度。

3. 抹除过程　它是将已记录的信息或残余的剩磁抹除的过程。采用交流磁场抹除是使媒体磁性粒子无序排列，磁媒体不再显示其磁性。而直流抹除则是使磁性粒子按同一方向排列，不出现磁化方向翻转，同样也读不出任何信号。通常，在磁记录中使用的是重写覆盖，无须先行抹除操作，只有为防止道间串扰才使用磁道边缘抹除。

4. 注意问题　读写过程中尚需注意以下问题。

（1）波形拥挤效应　前述读出过程中所讨论的只是单次磁化状态翻转的读出波形，忽略了相邻磁化状态翻转对读出波形的影响。任何数字磁记录的读出信号都是一连串的磁记录特殊波形，彼此间相互影响。当磁化状态翻转密度很高时，读出信号波形拥挤现象尤为显著。波形的拥挤导致幅度衰减与峰点偏移，它将影响数据与时钟的分离，严重时造成数码错误。幅度衰减主要因相邻波形极性相反相互削弱所造成，峰点偏移则因两侧的增强与削弱不平衡而引起。例如对于码字为 11100111 的读出波形，靠近二个 0 的 1，其峰点将向 0 位偏移。所以针对不同的编码，在写入时要进行补偿，即提前或滞后写入；或者在读出时进行读后均衡，以抑制峰点的偏移。

（2）波形不对称　波形不对称主要由磁化的垂直分量引起。波形是由水平分量（或称纵向分量）和垂直分量两部分合成的。水平分量对称于纵轴，而垂直分量则近似地对称于原点，而不对称于两个坐标轴，故合成后的波形不具备对称性。对于水平磁记录（或称纵向磁记录），由于媒体层厚极薄，磁化的垂直分量很小，故读出波形的垂直分量亦小，波形的不对称性并不严重。对于垂直磁记录，当以环形磁头读写时，垂直分量较大，读出波形具有明显的不对称性。为消除此种影响，可以采用电路纠正的方法或在磁头结构上采取措施达到读出波形对称的目的。

数字磁记录类型

1. 纵向磁记录　采用水平磁化场对平面内沿磁道方向取向的媒体进行磁化的磁记录称为纵向磁记录。水平磁记录使用环形磁头（或环形磁头写入、磁变阻磁头读出）与平面内取向的薄膜媒体。由于环形头的磁头场主要是水平分量，媒体极薄，且为平面内取向，垂直于媒体表面的退磁场很大，故记录后读出波形的对称性很好。它的翻转密度受各种因素的影响，可使用翻转过渡区参数来衡量。理论上，过渡区参数 a 可表示为

$$a = 2b(2-s)M_r/H_c$$

式中，b 为媒体厚度，s 为媒体磁滞回线的矩形比，M_r

是媒体剩余磁化强度,H_c 是媒体的矫顽力。若要提高翻转密度,便需减小过渡区参数 a。从上式可知,需要减小磁层厚度,提高媒体材料的矩形比和减小比值 M_r/H_c。

2. 垂直磁记录　采用易轴(easy axis)垂直于表面的各向异性媒体和垂直磁场进行记录的方式称为垂直磁记录。目前,绝大部分磁面存储器都采用垂直记录。与前述纵向磁记录相比,它有如下特点。

①记录后,磁化单元垂直媒体表面(如图 4 所示),且呈相互吸引结构,磁化状态稳定,翻转密度很高。②适宜于用单极型磁头读写,但也能用环形头记录。当使用环形头记录时,波形严重不对称,必须进行特殊处理。无论使用何种磁头,都要求磁头与媒体表面间的距离极小。③使用垂直表面取向的媒体,这种媒体有溅射 CoCr/Ni-Fe 的双层薄膜和涂布的钡铁氧体膜等。

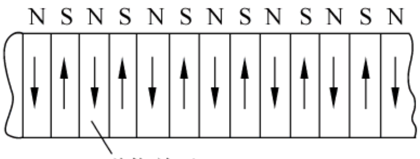


图 4　垂直记录磁化翻转图形

垂直磁记录技术已经广泛运用于**硬磁盘驱动器**中。预期在不久的将来,热磁辅助技术、微波辅助技术以及 patterned media 将成为推动磁记录技术发展的强有力手段。

参考文献

1. 张江陵,季国钧,等. 外部设备设计原理. 武汉:华中理工大学出版社,1989

2. 杨正. 磁记录物理. 兰州:兰州大学出版社,1986

(张江陵)

shuzi duoyongtu guangdie

数字多用途光碟(digital versatile disc, DVD)　在**数字可写光碟**(CD-R)基础上通过减短半导体激光器的波长(从 780 nm 减到 650 nm)、增大物镜的数值孔径,又称镜口率(从 0.45 增到 0.60)等技术手段,从而提高存储密度的高密度光碟系统。它是 CD 光碟的更新换代产品,系统由 DVD 光碟机和 DVD 光碟两部分组成,简称 **DVD 光碟**或 DVD。

DVD 光碟由美国时代华纳公司提出构想,日本东芝公司研制,开始称为超高密度数字视频光碟(SD);接着,以荷兰飞利浦公司、日本索尼公司为首研制出多媒体数字光碟(MMCD),因而形成两大阵营。1995 年 12 月,两大阵营在多次协商的基础上统一定名为 DVD。1996 年推出了 DVD 的第一个产品——DVD 视频播放机。

分类　DVD 光碟机根据用途的不同可分为 DVD 光碟驱动器、DVD 光碟播放机和 DVD 光碟录像机等。DVD 光碟驱动器用作计算机外部设备,对 DVD 或 CD 光碟进行信息的读取、写入、擦除。光碟驱动器根据其使用的 DVD 或 CD 光碟类型不同,分为只读、一次写及可擦写三类光碟驱动器。DVD 光碟播放机是只能回放 DVD-Video, DVD-Audio 等的信息家电产品。DVD 光碟录像机既可以是一次写的,也可以是可擦写的,并具有播放机功能。

DVD 光碟虽与 CD 光碟外型尺寸完全一致,但它是由两张 0.6 mm 厚的碟片粘接而成。DVD 光碟根据功能不同可以分为只读、一次写、可擦写和复合四种。只读 DVD 光碟有 DVD-ROM 、DVD-Video 和 DVD-Audio。DVD-ROM 用于存储计算机读取的信息。一次写 DVD 光碟有 DVD-R, DVD + R。可擦写 DVD 光碟有 DVD-RW, DVD-RAM 和 DVD + RW。DVD-RW 可重写 1000 次以上,DVD-RAM 是可随机存取的光碟,可重写 10 000 次以上。DVD + R 和 DVD + RW 分别基于飞利浦公司提供的一次写和可擦写光碟技术格式,未得到 DVD 论坛的认可。复合 DVD 光碟不是新的格式和标准,而是已有不同 DVD 格式的组合,从而与不同 DVD 格式最大程度地兼容。

DVD 光碟根据其物理结构可分为单面单层、单面双层、双面单层和双面双层四种规格,其中每种规格根据盘片直径又可分为 120 mm 和 80 mm 两种,最常用的是 120 mm。只读 DVD 光碟具有上述四种规格,其余均为单面单层。120 mm 单面单层只读 DVD 光碟容量为 4.7 GB(称为 DVD 5),单面双层容量为 8.5 GB(称为 DVD 9),双面单层容量为 9.4 GB(称为 DVD 10),双面双层容量为 17 GB(称为 DVD 18)。由于工艺过于复杂,一般不制作 DVD 18 光碟。

硬件结构　DVD 光学头是 DVD 光碟机的核心部件,由半导体激光器、物镜、光电探测器和执行器等相关光学及机电元件组成,实现信息的读出、写入及擦除。

DVD 伺服系统实现光学头读写光斑在光碟信

息道的聚焦、循迹和扫描功能，包括聚焦伺服系统、循迹伺服系统、寻址伺服系统和主轴伺服系统。聚焦伺服系统实现光斑的轴向定位控制，循迹伺服系统实现光斑的近距离径向定位控制，寻址伺服系统实现光斑的长行程径向定位控制，主轴伺服系统通过控制光碟的转速实现光斑对光碟信道的恒线速或恒角速等的扫描。

DVD 解码系统实现信道解码和音视频解码功能。除 DVD 光学头之外，伺服和解码均由专用芯片实现。

DVD 技术在提高数据存储密度和数据传输速率两个方向发展。蓝光光碟技术可以将单面单层盘片容量提高到 25～30 GB，多波长多阶光盘存储、全息存储和宽光束存储等新技术均可显著提高存储密度和数据传输速率。

参考文献

徐端颐. 光盘存储系统设计原理. 北京：国防工业出版社，2000

（潘龙法）

shuzi huayin de yasuo bianma

数字话音的压缩编码（compression and coding of digital voice） 对人们说话产生的话音数字化之后进行数据压缩和编码的方法与技术。话音是人们说话时肺部的空气通过声门沿着声道发出时产生的，其信号带宽比乐音窄。例如电话话音信号的带宽只有 300～3400 Hz，取样频率一般为8 kHz，量化精度为 8 位，每秒钟的数据量（码率）是 64 kb（8 kB），1 小时的数据量大约是 28 MB（字节）。为了降低存储成本和提高通信效率（降低传输带宽），对数字话音进行数据压缩是十分必要的。

由于声音信号中包含有大量的冗余信息，再加上还可以利用人的听觉感知特性和话音信号的生成机理，因此，大幅度压缩数字话音的数据是完全可能的。目前，数字话音的数据压缩方法很多，从原理上大致可以分成如下 3 类。

（1）波形编码　这是一种基于话音数据的统计特性的压缩方法，例如 PCM（脉冲编码调制）、ADPCM（自适应差分脉冲编码调制）、子带编码等。它们的码率虽然比较高（分别为 64 kb/s 和 32 kb/s），但能保证话音的高质量，且算法简单、易实现，在电话中继通信中得到广泛应用，如 CCITT G. 711 和 G. 721。此类方法通用性好，不仅适用于数字话音的压缩，而且对所有波形形式的数字声音都有效。不足之处是难以达到很高的压缩比。

（2）模型编码　又称为参数编码或源编码。这是一种基于话音生成模型的压缩方法，它使用“声源-滤波器”模型来模拟人的发声过程。压缩编码时，使用线性预测方法从原始的话音波形信号中提取话音的生成参数，并把这些参数作为原始话音压缩编码的结果，因此码率很低。解码时必须使用与编码完全相同的模型，所以可应用于保密通信，例如声码器。此类方法的优点是压缩比很高，但质量还不很理想。

（3）混合编码　又称“合成-分析法”（AbS），这是上述两种方法的结合。它从原始的话音波形信号提取“声源-滤波器”模型中的声道参数与激励信号，通过优化激励信号编码和感知加权滤波，设法使采用这些激励信号所产生的波形尽可能与原始话音的波形一致。采用此类压缩编码方法后，数字话音的码率一般为 4. 8～16 kb/s，达到了较高的压缩比，同时又能保证较好的话音质量。其中使用最多的是码激励线性预测法（CELP），目前移动通信和 IP 电话中话音信号的压缩编码大多采用这种方法。

图 1 是上述 3 类数字话音压缩编码方法的比较。

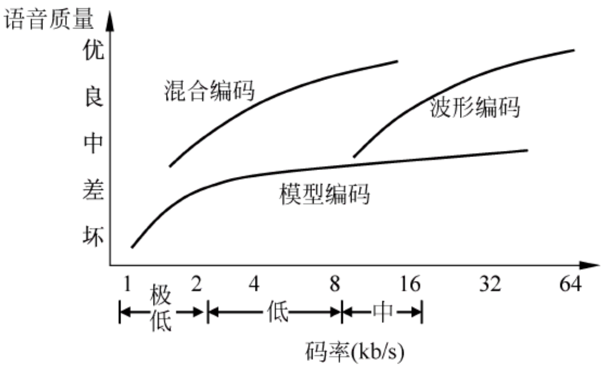


图 1　数字话音压缩编码方法的比较

参考文献

1. Sanjeev Gupta. Science of speech coding. DRDO Science Spectrum, 2009: 120-127

2. Kondoz A M. Digital speech coding for low bit rate communication systems. John Wiley & Sons, 1999

（张福炎）

shuzi jihe chuli

数字几何处理（digital geometry processing） 以数字化的几何数据及相关的附着物理属性为对象，部分借鉴传统信号处理方法，按特定目的对数据实施去噪、压缩、转换、增强、检测和分析等操作。广

义的数字几何处理流程包含以下阶段：①创建和获取；②存储与传输；③认证；④编辑与动画；⑤仿真计算；⑥加工制造等。

数字几何数据是一种不同于图像、音频和视频的新数据媒体。随着三维信息获取技术的不断成熟，三维几何模型在影视娱乐、设计与制造、**电子商务**、模拟和仿真等领域得到了广泛的应用，从而推动了数字几何处理技术的产生和发展。

在处理的不同阶段，数字几何数据采用了多种表示方法，根据数据的表述形式，可概要区分为连续和离散两种数据类型。离散数据，包括体素、多边形网格和**点云**数据，是通过一定方式对物体的几何信息进行采样获得。该类数据往往出现在整个处理的早期，即数据获取阶段。由于多边形网格数据的结构简单，利于绘制和表示复杂形体，适于与其他数据类型转换，因此其应用贯穿了整个数字几何处理过程，成为主要研究内容（详见**三维网格处理**）。连续数据采用显式或者隐式的连续数学解析表示，其优点在于表示紧凑，利于给出任意精度的采样。再者，如**参数曲面**等显式表示，可快速赋值，适于直接的曲面形状交互和控制。而以径向基函数曲面为代表的**隐式曲面**则便于求交计算，具有良好的插值和外推能力，可用于填补获取数据的缺失部分，避免曲面的局部重叠。目前，研究者还发展出了多种过程式和混合式的表示处理方法，如结合参数曲面和多边形网格优点的细分方法，结合空间八叉树**细分曲面**生成技术来提高赋值和查询速度等。

几何数据上附着的物理属性主要指物体的外观描述，在计算机中常被描述为颜色、纹理、光泽和透明度。由于真实世界的物体表面细节极其复杂，人们研究了多种经验和物理模型，如双向反射函数、纹理映射、凹凸和位移纹理，以及双向反射纹理等。在数字几何处理中，这些附着属性与三维几何位置信息一样，一般关联于对应顶点，因而可通过增加顶点信息的维数而纳入原有的几何处理框架。

不同于其他 3 种数字媒体的处理，由于其数据特性，使得数字几何处理对目前的计算框架提出了重大挑战。主要的难点有：①获取的离散数据往往是非均匀采样，噪声产生原因复杂，存在数据残缺现象；②数据的几何和拓扑依赖性强，不利于使用原有的信号处理框架；③数据量变化剧烈，海量数据在未经处理时往往超过目前计算机的处理和显示能力，需考虑合理分片、简化或者外存（out-of-core）技术。

近年来，随着网格变形、参数化问题的深入研究，使得人们对数字几何处理有了新的理解，产生了新的理论和计算方法。其中比较有代表性的是，部分研究者研究了数字几何数据的离散微分属性，进而发展出了一系列微分域求解技术和形状操纵方法。这些方法的核心是通过计算与操纵数字几何数据的离散微分属性，在保持局部几何细节的同时做到整体形状变形。

参考文献

1. Schröder P, Sweldens W. Digital geometry processing. ACM SIGGRAPH 2001 Course, Los Angeles, California, August, 2001

2. 周昆. 数字几何处理：理论和应用. 浙江大学博士论文，2002

3. Desbrun M, Schröder P, Wardetzky M. Discrete differential geometry: an applied introduction. SIGGRAPH Asia 2008 Course, 2008

（鲍虎军　张宏鑫）

shuzi jisuanji

数字计算机（digital computer）　使用离散符号或数字表示数据、在程序控制下自动进行计算的电子装置，其全名是电子数字计算机。由于这类电子计算机已经普遍应用于人类社会的方方面面，通常人们提到的计算机往往就是指数字计算机。

数字计算机包括硬件和软件两大部分。硬件是组成计算机的电子器件及其线路、部件和外围设备等的统称（参见**计算机硬件**）；软件则是使计算机能够自动运行以完成信息处理任务的各种程序及其文档的统称（参见**计算机软件**）。下面扼要介绍计算机核心结构的组成和工作原理。

计算机硬件系统的组成

计算机的硬件系统主要由中央处理器、主存储器和外围设备组成。当然，支持和保障计算机得以正常运行的机箱、电源、通风散热设施等也是必不可少的，因它们不涉及计算机的工作原理，以下不再提起。下面将分别介绍各组成部分的主要功能。

（1）中央处理器（CPU）　主要由控制器、运算器、内部寄存器组和处理机总线等部分组成（参见**中央处理器**）。控制器的主要功能是自动从主存储器读取指令，予以解释和执行，并控制与内部寄存器组、主存储器或外围设备交换信息和数据；运算器的主要功能是按照指令的要求完成相应的算术或逻辑运算；处理机总线的功能是将 CPU 内部的各部件连

接起来。

（2）主存储器（MM） 简称主存，亦称内存。它是可随机存取的操作存储器，所以也称**随机存储器**（RAM）（参见**主存储器**）。它存储计算机运行时随时所要用到的程序和数据。现代计算机采用快速大容量的**半导体存储器**。

（3）**外围设备** 输入输出设备和外存储器的统称。计算机运行时所需要的程序和数据都要由输入设备输入到内存中；计算机在进行信息处理的过程中所产生的最终结果和某些中间结果要通过输出设备送出；受内存容量的限制，加以半导体 RAM 中的信息在断电时会全部消失，所以需要有大容量的、能长期保存数据的外存储器。常用的输入装置是**键盘**和**显示器**。显示器同时也是输出显示装置，是操作员和用户与计算机相互联系的窗口。其他常用的输出装置有针式打印机、激光印刷机、喷墨印刷机、**扫描仪**、**绘图仪**、数码相机等。常用的外存储器是**磁盘存储器**和**磁带存储器**。磁盘又有软磁盘和硬磁盘之分。软磁盘和磁带因便于装卸，又可脱机保存数据，必要时也可兼作输入媒体用。移动硬盘的容量比软磁盘大，又可以通过 USB 接口方便插接，兼有较大的存储容量并且便于移动的优点。光盘能够不受电磁干扰，并能够长期保存大量信息。

计算机中信息的表示

计算机采用二进制编码方式表示数、字符、指令和其他控制信息，这不但比十进制表示简单，而且便于用最简单的开关电路实现二值逻辑运算。计算机在存储、传送或处理数据时，作为一个单元的一组二进制代码称为**字**，一个字中的二进制位的位数称为**字长**。常用的字长有 8 位、16 位、32 位、64 位。字长为 8 位的代码称为**字节**，它是计算机中表示符号或字符的基本编码单位。

数的表示（参见**数制**） 人们早已习惯用十进制来表示数，其要点是“逢十进一”，即用 0，1，…，9 十个值中的任一个值来表示一位十进制数，当一个数大于 9 时，就需用多位来表示。在十进制记数系统中，第一位为个位，其权值 $R_0=10^0=1$；第二位为十位，其权值 $R_1=10^1=10$；……即每进一位，其权值增加十倍。多位数的值等于各位数的值之和。十进制数 N_{10} 可表示为

$$N_{10}=\sum_{i=1}^{n} r_i\times 10^{i-1}$$

式中，n 为位数；$r_i=0,1,\cdots,9$。

例如，$1036=1\times10^3+0\times10^2+3\times10^1+6\times10^0=1000+0+30+6$。

相应地，二进制数 N_2 可表示为

$$N_2=\sum_{i=1}^{n} b_i\times 2^{i-1}$$

式中，$b_i=0,1$。

上述十进制数 1036 可用二进制表示为

$$\begin{aligned}N_2 &= 10000001100\\ &= 1\times2^{10}+0\times2^9+0\times2^8+0\times2^7+\\ &\quad 0\times2^6+0\times2^5+0\times2^4+1\times2^3+\\ &\quad 1\times2^2+0\times2^1+0\times2^0\\ &= 1024+8+4=1036\end{aligned}$$

对于负数，一般用数的最高位作为符号位。符号位为 0 表示正数，为 1 表示负数。符号位以外的数位为数的绝对值，这样的数码称为原码。除了用原码表示数外，还可用补码或反码表示数（参见**数制**）。

在计算机内直接表示一个整数或小数的方法称为**定点表示法**，因为小数点总是指定在某一固定位置而得名。另一种表示数的方法是**浮点表示法**。这是把一个字分为两段：前一段表示带符号的阶码 p，即表示 $2^{\pm p}$，p 为整数；后一段表示带符号的定点小数 m，称为尾数。用浮点表示法表示的数为：$N_2=\pm2^{\pm p}m$，符号位在最前面。显然浮点表示法所表示的数的范围比定点表示法的大得多。

在计算机中，有时用八进制或十六进制表示数，可分别用 3 位或 4 位二进制数来表示。一个十进制数也可用 4 位二进制数表示，这种数称为二–十进制数。用二进制数表示十进制数有多种编码形式，包括 8421 码、余 3 码、格雷码等。

字符的表示 国际通用的字符编码标准是 **ASCII 码**。用一个字节中的低七位编码表示 $2^7=128$ 个字符，最高位用于奇偶检验或置为 0。字符包括十进制数码，大、小写英文字母，标点符号，常用算符，各种控制符和特殊标志等。同样也可以用二进制数编码表示汉字。由于一个字节的 8 位编码最多只能表示 $2^8=256$ 个不同的字符，远远少于不同汉字的数量。所以，至少必须用两个字节来表示不同汉字的编码。按标准规定，令其中每个字节的最高位为“1”，以示与 ASCII 码的区别（参见**字符集**）。

指令的表示 指令是指示计算机操作的命令。通常用一个以上字节的编码来表示一条指令。指令主要由操作码、地址码等部分组成，它们所占用的位数，决定操作种类的数量和最大可访问的存储空间。例如，7 位操作码可以表示 $2^7=128$ 种不同的操作；

10 位地址码的寻址空间为 $2^{10}=1024$,称为 1K。16 位地址码的寻址空间为 2^{16},即 64K。此外还要设置若干位来表示指令的特征和寻址方式等。

数字和字符的编码表示统称为数据,它们是指令的操作对象。指令和数据同样用二进制编码形式表示,这是计算机得以在程序控制下完全自动进行数据处理的关键。

逻辑线路

二进制数位的两个值 0 或 1,正好与命题逻辑的“真”和“假”一一对应。于是,二进制的运算可以很自然地转化为**逻辑运算**。最基本的逻辑运算为:①“非”,记为 $\bar{A}$ 或 NOT A。即,若 $A=1$,则 NOT $A=0$;反之亦然。②“或”,记为 $A \vee B \vee \cdots$ 或 A OR B OR $\cdots$,表示命题 $A, B, \cdots$ 中只要有一个或一个以上命题为真,其结果即为真。相应的布尔表达式为 $Z=A+B+\cdots$。③“与”,记为 $A \wedge B \wedge \cdots$ 或 A AND B AND $\cdots$,表示当且仅当全部命题 $A, B, \cdots$ 皆为真,其结果方为真。相应的布尔表达式为 $Z=AB\cdots$ 或 $Z=A \times B \times \cdots$。由于二进制数只有 0 和 1 两个值,所以,可用电子开关电路,或称“门”电路来实现相应的运算。与上述 3 种基本的逻辑运算相对应的基本逻辑线路是“非门”、“或门”和“与门”,它们可以组成计算机各种不同功能的逻辑线路。例如,用两个“或非门”或两个“与非门”可以组成一个触发器,可作为一位二进制码的存储单元;多个触发器并列且与相应的输入、输出控制门连接,可组成能存储一个字的寄存器;还可以用基本逻辑电路组成**加法器**、译码器、编码器、时序信号发生器、分频器、移位寄存器等。前述计算机的各主要部件,如 CPU,MM,外围接口等极其复杂的线路,也都是由简单的基本逻辑电路加上一些辅助电路所组成。

存储程序概念

计算机 ENIAC 开创了电子计算机发展的新纪元。但是,控制其运算的程序却是在类似于模拟计算机的排题板上,用外接线接插编排的。早在 ENIAC 的研制过程中,其主设计师 J P Eckert 和 J W Mauchly 就有了用超声波汞延迟线作为二进制代码的串行存储器来动态保存指令的想法,并在美籍匈牙利数学家冯·诺依曼的参加下,着手制定了重新研制一台能把程序和数据一同存储在操作存储器中,从而实现在存储程序控制下自动计算的计算机 EDVAC 的计划。冯·诺依曼在由他执笔的“关于 EDVAC 设计方案的初步报告”(1945 年 6 月 30 日)中,首次明确地阐述了存储程序概念的基本设计思想。冯·诺依曼等在 1946 年发表的题为“电子计算装置逻辑设计的初步讨论”的论文中,进一步系统而深入地阐述了以存储程序概念为指导的计算机逻辑设计思想,勾画出一个完整的计算机体系结构,至今仍不失其开创性的指导意义。后来人们称这种体系结构为冯·诺依曼结构;称采取这种体系结构的计算机为*冯·诺依曼计算机*。EDVAC 虽然于 1945 年开始研制,但由于种种原因,到 1951 年才正式投入运行。

冯·诺依曼结构的核心是存储程序概念。其要点为:①指令与数据均储存在主存储器中;②在 CPU 的控制器中,设置一个**程序计数器** PC,其初始值置为程序的第一条指令所在存储单元的地址。计算机一旦被启动,控制器就会自动按 PC 的当前值所指,从主存储器读出指令,予以解释、执行。同时,PC 自动加 1,指向顺序执行的下一条指令。执行每条指令所涉及的操作数,则按指令中的操作数地址所指,从主存储器读取或向主存储器写入。以上要点如图 1 所示。要执行的程序和有关数据一同存放在内存(即主存储器)的不同区间。开始执行程序时,其首地址放在 CPU 中特设的程序计数器(PC)中。按 PC 中的首地址读出第一条指令,传送到指令寄存器(①,②),同时 PC 自动加 1,准备好读取顺序存放的下一条指令。执行指令所涉及的数据则按指令寄存器中的有效地址码所指的存储单元,进行读出或写入(③,④)。值得一提的是,某些指令(例如转移指令等)中的地址可以是另一条指令或另一段程序(例如子程序、中断服务程序、另一个分时运行的程序等)的指令所在存储单元的地址,这就增加了自动调度程序的灵活性。现代计算机正是这样得以实现在程序控制下全自动地进行信息处理。

计算机的程序设计语言

由图 1 可见,在所执行的程序代码已经存放在内存的前提下,计算机的工作过程就是从读出第一条指令开始,顺序读出、解释和执行每一条指令的过程。这些用二进制编码表达的指令构成了计算机能直接“理解”并严格执行的代码语言,称为**机器语言**。这种语言与人类的自然语言相差很远,直接用它来编写程序是一项十分费时、难懂且很容易出错的工作。因此,早在计算机发展初期的 1953 年,就开始采用所谓“助记符”的符号来书写指令,从而产生了**汇编语言**。用汇编语言编写的程序,先要通过**汇编程序**转换为代码程序(亦称目标代码),再由计算机来运行。为了进一步从根本上克服用机器语言

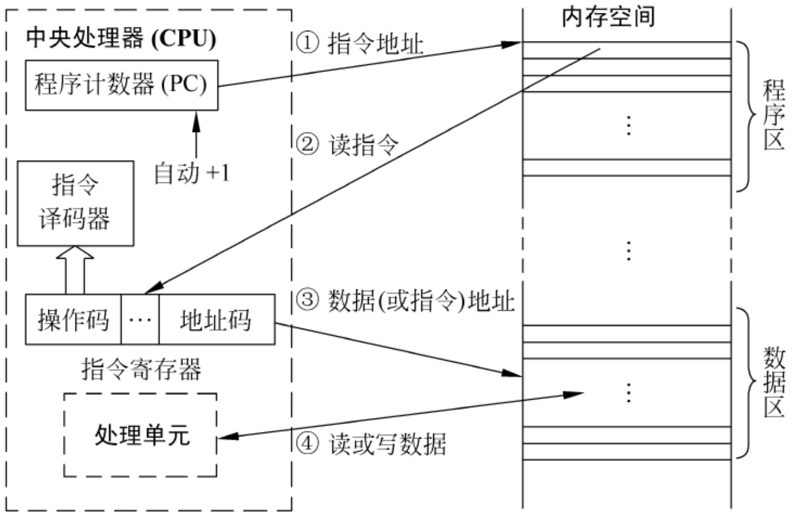


图 1　存储程序控制的原理

编程所带来的种种弊端,Backus 在 1957 年首先提出第一个高级程序设计语言 FORTRAN,并设计了相应的**编译程序**(或称编译器)。这种编译程序使人们得以摆脱计算机指令系统和机器语言的约束,可以用易写和易懂的形式语言来编写程序。此后,各种**高级语言**相继出现,包括 ALGOL-60,BASIC、COBOL,PASCAL,C,C++,LISP,Ada,PROLOG,Java 等。

参考文献

1. Hamacher C, Vranesic Z, Zaky S. 计算机组成. 5 版. 影印版. 北京: 机械工业出版社,2002

2. 王爱英. 计算机组成与结构. 5 版. 北京: 清华大学出版社,2013

(童频　刘志勇)

shuzi kechongxie guangdie

数字可重写光碟(compact disc-rewritable, CD-RW)　采用相变材料作为存储介质并可以多次重写的数字光碟。主要的品种有 **CD-RW** 和 **DVD-RW/DVD+RW**。

为了满足众多应用领域对 CD 光碟提出的可多次重复擦写的要求,Philips 公司在 1995 年 4 月制定出了与光碟只读存储器(CD-ROM)和**数字可写光碟**(CD-R)兼容的相变型可擦写光碟驱动器 CD-E(CD erasable)标准。1996 年 10 月,Philip、Ricoh、Sony、HP 和 Mitsubish 五家公司在 CD-E 的基础上联合制定了新的可擦写 CD 标准,并将 CD-E 更名为 CD-RW(CD-ReWritable)。1996 年底,Philip 和 Ricoh 分别向全世界推出了第一台 CD-RW 可擦写刻录机和第一张 CD-RW 可重写光碟。CD-RW 可擦写刻录机不仅能读取 CD-ROM 光碟、刻录 CD-R 光碟,而且还能对专门的 CD-RW 光碟擦写多次。

DVD-RW 的全称为 DVD-ReWritable(可重写式数字多用途光碟),不过业界为了将其与 DVD+RW 区分,定义为 Re-recordable DVD(可重记录型 DVD)。DVD-RW 标准是由 Pioneer 公司于 1998 年提出的,并于 2000 年中完成 1.1 版本的正式标准,标准容量为 4.7 GB。DVD-RW 产品最初定位于消费类电子产品,然而随着技术发展,DVD-RW 的功能也扩充到了计算机领域,其 1.1 版支持 CPRM 版权保护技术。

DVD+RW,是另一种可重写的 DVD 格式光碟,这种格式由 Philip 等公司组成的 DVD+RW 联盟开发,最早的容量仅有 2.8 GB,至 2001 年,DVD+RW 容量提升到 4.7 GB。DVD-RW 和 DVD+RW 的主要区别在寻址方式,DVD-RW 使用低频(140.6 kHz)的摆动沟槽(wobble),寻址时依靠凸轨处预刻的寻址信息坑;DVD+RW 使用高频(817.4 kHz)摆动沟槽,寻址利用在预刻凹轨处摆动沟槽的相位调变。DVD+RW 支持恒角速度方式。由于是不同的标准组织所制定的标准,相互之间并不兼容,目前绝大多数 DVD 光驱/刻录机和影碟机同时能兼容 DVD-RW 光盘和 DVD+RW 光碟。

CD-RW/DVD±RW 碟片的结构与原理

CD-RW 碟片与 CD-ROM 碟片很相似,DVD±RW 碟片与 DVD-ROM 碟片很相似,它们的外形尺寸都一样,其直径为 120 mm,厚度约为 1.2 mm。CD-RW/DVD±RW 碟片主要由 6 层组成,分别是碟基、下绝缘层、记录层、上绝缘层、光反射层和保护

层。为了达到多次写入的目的,CD-RW/DVD ± RW 使用了相变技术。它通过一种可逆变结晶材料(通常是稀有金属合金)在结晶与非结晶状态下不同的光特性来记录数据,而这两种状态可以在高/中功率激光的照射下相互转换,因而可删除并重新写入数据,记录层即是用这种材料制成。

初次使用的 CD-RW/DVD ± RW 碟片的记录层处于结晶状态。在进行刻录时,首先按数据中的 0/1 信号的转换来调制刻录激光的照射开关。此时的激光是短时、高功率的,被照射记录层的温度会迅速超过熔点,而在迅速冷却后就变为非结晶状态。此时,结晶的地方可让光线通过到达反射层,没有结晶的地方则很难让光线通过,从而造成了激光反射功率上的差异。在进行数据抹除时,用中等功率的激光对非结晶状态的记录层进行相对长时间的照射,使之慢慢达到结晶的温度,从而转换回原来的结晶状态。由于记录层使用稀有金属合金制成,所以为了能让记录层的结晶状态得以长期保持,又在它的两边铺上了上绝缘层与下绝缘层来杜绝其与外界电磁方面的联系。另外, CD-RW/DVD ± RW 碟片在出厂时也已刻好了预刻槽。

CD-RW/DVD ± RW 碟片可以反复使用 1000 多次。由于多了好几层物质的阻挡,CD-RW 碟片的光反射率只有 15% ~25% ,DVD ± RW 碟片的反射率为 18% ~30% 。

DVD ± RW 碟片都有相应的单面双层规格(DVD ± RW DL),其容量为 8.5 GB,不过由于稳定性问题,目前市场上还是以单面单层的碟片为主。

在刻录时,可以将 CD-RW/DVD ± RW 碟片当作 CD-R/DVD ± R 碟片来使用,但由于 CD-RW/DVD ± RW 允许擦写,所以一般都将它当作一个硬盘来用,此时它就好像是一个新的分区,但是必须要先用软件对碟片进行格式化。

DVD 刻录机

早期的刻录机主要是 CD-RW 刻录机以及将 CD-RW 和 DVD-ROM 技术相结合的 Comb 机种,由于技术进步很快,目前已经是以 DVD 刻录机为主。一般的 DVD 刻录机都可以兼容 CD-RW/DVD ± RW 碟片,可反复读写碟片上的数据,使在光碟上刻录数据如同在磁盘上存储数据一样容易。

DVD 刻录机的一个重要指标是缓存的大小,缓存的大小将直接影响到刻录的成功率。据调查统计,至少 80% 的刻录失败都与缓存容量不足有关。现在的 DVD 刻录机一般使用 4 MB 缓存并开始向 8 MB 发展。由于在刻录光碟时需要光碟稳定、均匀地旋转,因此对伺服系统的精度、可靠性与寿命都提出了很高的要求,DVD 刻录机的伺服系统要普遍好于 DVD-ROM 驱动器。

DVD 刻录机的另一个重要指标就是刻录速度,目前对 DVD ± RW 碟片的刻录速度已经到 8 X 倍速和 16 X 倍速,对 CD-RW 碟片的刻录速度已经达到 32 X 倍速和 48 X 倍速(1 X = 150 KB/s)。另外,写入速度与碟片的速度限制是密切相关的。这是因为要想加快写入速度,碟片的旋转速度就要提高,这将使单位面积上碟片所接收到的激光功率减少,虽然高速驱动器会加大激光的能量输出,但碟片所用介质的灵敏度也要提高。现在的 CD-RW/DVD ± RW 碟片都会在其商标一面注明最高速度限制(如 4 X、8 X 等)。

参考文献

1. 徐端颐,等. 高密度光盘数据存储. 北京:清华大学出版社,2004

2. 干福熹,等. 数字光盘存储技术. 北京:科学技术出版社,1998

(黄浩 谢长生)

shuzi kexie guangdie

数字可写光碟(compact disc-recordable, CD-R) 采用可重写一次记录材料作为存储介质的数字光碟。主要有 CD-R 和 DVD-R/DVD + R。CD-R 的另一英文名称是 CD-WO(write once),就是只允许写一次,写完以后,记录在 CD-R 碟上的信息无法被改写,但可以在多种类型驱动器如:CD-ROM 驱动器、DVD-ROM 驱动器、CD-R 驱动器和 CD-RW 驱动器上被反复地读取多次(参见**只读光碟驱动器**)。

DVD-R 的全称为 DVD-recordable, 1997 年日本 Pioneer 公司发布了全球第一款 DVD 刻录机,使用的碟片是 DVD-R,这是第一种可用的 DVD 可记录格式,标准由 DVD Forum 定义。该碟片按照 Pioneer 公司的 DVD-R 规格 1.0 版设计,当时设计存储能力为 3.95 GB。1999 年底,Pioneer 公司发布 DVD-R 规格2.0 版,设计存储容量为 4.7 GB。

DVD + R 是另一种可记录的 DVD 格式光碟,2001 年飞利浦公司发布了 DVD + R 光碟,设计存储容量为 4.7 GB。DVD-R 和 DVD + R 的主要区别在寻址方式,DVD-R 使用低频(140.6kHz)的摆动沟槽(wobble),寻址时依靠凸轨处预刻的寻址信息坑;DVD + R 使用高频(817.4kHz)的摆动沟槽,寻址时

利用在预刻凹轨处摆动沟槽的相位调变。由于是不同的标准组织所制定的标准，两种格式互不兼容，目前绝大多数 DVD 光驱/刻录机和影碟机同时能兼容性 DVD-R 和 DVD + R 两种格式的光碟。

CD/DVD ± R 碟片的结构与记录原理

CD-R 碟片与 CD-ROM 碟片有许多共同之处，外形尺寸完全一样，其直径为 120 mm，厚度约为 1.2 mm。与 CD-ROM 碟片的三层结构所不同的是，CD-R 碟片采用了四层结构形式，底层为聚碳酸酯透明塑料注塑成形的衬盘，在塑料衬底上镀有一层很薄的有机染料记录层，并使用抗腐蚀的金属膜做反射层，使 CD-R 碟片达到 70% 的反射率，反射层外为涂漆保护层。当写入激光束聚焦到记录层上时，染料被加热后烧熔，形成一系列代表信息的凹坑。这些凹坑与 CD-ROM 碟片上的凹坑类似。有些 CD-R 碟片在漆保护层之上还用吸墨材料涂有第五层印刷层，用户可用喷墨打印机直接在 CD-R 碟片背面打印商标、图案或文字说明，也可用软笔进行标注。

刻录 CD-R 碟片时，CD-R 碟片以恒定线速度旋转，功率可调的激光束沿着螺旋形预刻槽烧出一系列相间的不同长度尺寸的微孔，以实现记录数据的正确写入。由于有机染料记录层的微孔处具有强的光反射特性，而未溶解汽化的有机染料反射率很低，因此 CD-R 碟片上的微孔与未熔解的有机染料分别具有与 CD-ROM 碟片上的平面和凹坑相同的光学反射特性，可以利用 CD-ROM 驱动器和 DVD-ROM 驱动器进行数据读取。

DVD-R/DVD + R 碟片与 DVD-ROM 很相似，碟片的外形尺寸完全一样，其直径为 120 mm，厚度约为 1.2 mm。DVD-R/DVD + R 碟片也采用了四层结构形式，DVD-R/DVD + R 碟片记录原理与 CD-R 相同，DVD-R/DVD + R 碟片记录层采用稳定化金属氰兰染料，DVD-R/DVD + R 碟片的反射率为 45% ~ 85%。当几毫瓦的激光束聚焦到染料层上时，染料层被烧成气泡和斑痕，它与原染料层的反射率就不相同，因而可用来记录“1”和“0”两种数字信号。烧成以后就不可恢复，因此只能做一次性写入用，但写好后可以反复读取。DVD-R/DVD + R 碟片刻录方式与 CD-R 类似，DVD + R 支持恒角速度方式以及追加方式记录。

DVD ± R 碟片都有相应的单面双层规格（DVD ± RW DL），其容量为 8.5 GB，不过由于稳定性问题，目前市场上还是以单面单层的碟片为主。

CD-R/DVD ± R 刻录机

目前市面上的 CD-R 空白碟片主要有绿盘、金盘和蓝盘三种类型。这三种碟片的主要区别在于，它们分别使用了花青（cyanine）、酞花青（phthalocyanine）和金属化偶氮（AZO）化合物三种不同颜色的有机染料，从而使 CD-R 碟片呈现出绿、金、蓝三种不同的颜色。DVD ± R 碟片也有各种颜色，这主要是制造商在染料中增加了颜色添加剂造成的。从数据记录和读取的原理来看，不同颜色的 CD-R/DVD ± R 碟片都具有相同的功能。

CD-R 刻录机不仅可对 CD-R 空白碟片进行刻写，而且能读取普通的 CD-ROM，写完后的 CD-R 碟片可在 CD-ROM 驱动器、DVD-ROM 驱动器、CD-R 驱动器和 CD-RW 驱动器上被多次读取。CD-R 刻录机的刻录速度经历了低倍速到高倍速的发展阶段。目前，由于刻录机的快速发展，单独的 CD-R 刻录机已经逐渐被 DVD 刻录机取代。

DVD ± R 碟片可以用专门的 DVD ± R 刻录机来刻录，最大速度已经达到 20X 倍速（1X = 150 KB/s）；也可以用兼容 DVD ± R/DVD ± RW/DVD-RAM 的 DVD 刻录机来刻录，大部分 DVD 刻录机都支持 DVD ± R 碟片。

CD-R/DVD ± R 碟片只能写一次的缺点从另一方面来看却是一种优点。由于 CD-R/DVD ± R 碟片无法被改写，因此 CD-R/DVD ± R 碟片具有极高的安全性。CD-R/DVD ± R 碟片为那些需要永久性存储信息而又不准任何人擦除或更改的用户提供了一种最佳的数据存储解决方案。

参考文献

1. 徐端颐，等，高密度光盘数据存储. 北京：清华大学出版社，2004

2. 干福熹，等，数字光盘存储技术. 北京：科学技术出版社，1998

（黄浩　谢长生）

shuzi shexiangtou

数字摄像头（digital video camera）　计算机的一种数字视频输入设备。它采用镜头、光圈等光学系统聚焦成像，通过图像传感器将光学信号转换为模拟图像电信号，内部电路再将模拟图像信号转换为数字图像信号，并通过互连接口或网络传送到计算机。

图像传感器是组成数字摄像头的主要组成部分，根据元件不同分为电荷耦合器件（CCD，charge coupled device）和互补金属-氧化物-半导体（CMOS，complementary metal-oxide semiconductor）两种类型。

两种传感器的差别体现在图像质量和性能上，其中CCD传感器电路比较复杂，但具有较高的图像质量和性能；CMOS传感器的制造成本和功耗都低于CCD，通过采用影像光源自动增益补偿技术，自动亮度、白平衡控制技术，色彩饱和度、对比度、边缘增强以及伽马矫正等先进的影像控制技术，也可以达到与CCD传感器相媲美的效果。

数字摄像头与计算机的连接一般通过通用串行总线（USB）接口。数字摄像头的功耗一般较小，依靠USB提供的电源就可以工作，因而不需要另接电源。有些数字摄像头还具有网络接口，通过网络将图像传送到计算机。

数字摄像头的主要参数如下：

（1）分辨率　数字图像分辨率是指整个图像画面垂直和水平方向像素数的乘积。摄像头分辨率是摄像头所拍摄的静态或动态图像所能达到的分辨率。例如1920×1080，1280×720，640×480，352×288，320×240，176×144，160×120等。另外，通过软件插值放大，能够在上述原始的光学分辨率的基础上达到更高的图像分辨率，使图像、影像表现出丰富的细节，以达到真实的效果。

（2）像素数　一帧图像中包含的像素的个数，它是由图像传感器上的光敏元件数目所决定的，一个光敏元件就对应一个像素。市场上一些产品标称的像素数通常是指借助软件增强后的像素数。一般来说，像素越高的产品其图像的品质越好。但另一方面，像素越高，其记录的数据量也必然越大，对于存储设备和通信传输的带宽要求也越高。

（3）色彩位数　反映摄像头能正确记录颜色种类的多少，又称位深度。色彩位数的值越高，越能真实地表现亮部及暗部的色彩细节。

（4）最大帧数　每秒钟捕获的最大视频帧数，又称视频速度。在一定的数据传输率下，视频速度和分辨率呈反比关系。一般情况下，视频速度要达到30帧/秒，才能确保画面的连续性。如果数据传输率受限，提高分辨率往往会降低视频速度，造成画面跳动的现象。

数字摄像头作为一种计算机辅助配件，一般还配有专用的控制程序，作为用户直接控制和使用摄像头的工具软件，实现拍照、摄像、设置和管理影像文件的基本功能。（鲁宏伟）

shuzi shipin bianji

数字视频编辑（editing digital video）　采用专门的软件与硬件对数字视频进行各种编辑处理以达到使用要求的过程和技术。

数字视频的编辑处理，不再像模拟视频那样借助磁带录放像机进行，而是在大容量的可以随机存取的磁盘存储器上进行。它摆脱了按磁带顺序存取的束缚，使视频节目的制作效率得到了极大的提高。由于不受时间顺序限制，它们被称为非线性编辑系统。非线性编辑系统一般由计算机主机、视（音）频卡、SCSI硬盘、视频编辑软件和一些控制装置组成。非线性编辑系统的核心软件是数字视频编辑器。编辑时把电视节目素材存入计算机硬盘中，根据需要对不同长短、不同顺序的素材进行剪辑，同时配上字幕、特技和各种动画，再进行配音、配乐，最终制作成高质量的电视节目。

非线性编辑系统具有诸多优点，主要有四点。第一，编辑制作方便。非线性编辑可以先编好镜头的顺序，然后根据要求在需要的编辑点添加特技。第二，有利于反复编辑和修改。可在任一编辑点插入一段素材，切入点以后的素材可自动向后退，同样删除一段素材，切出点以后的素材可以自动向前递补，重组素材段。所有这些操作可以在几秒钟内完成。第三，制作图像画面的层次多，每一段素材都相当于传统编辑系统中一台播放机播放的视频信号，而素材数量是无限的，这使得节目编辑中的连续特技可一次完成无限多个。它不仅提高了编辑效率而且丰富了画面的效果。最后，图像与声音的同步对位准确方便。图像通过加帧减帧可以拉长或缩短镜头片段，声音可不变音调而改变音长（即保持声音频率不变，延长或缩短时间节奏）。

非线性编辑系统实际上是一种视频处理功能特强的多媒体计算机系统，借助于各种软件，它还兼具数字特技机、字幕机、编辑机、调音台的众多功能。以数字视频特技为例，它可以完成多种二维和三维特技控制，例如淡入淡出、变焦、推变等过渡功能，几十种图案划像功能，图像转换功能，图像加边功能，各种图像校正功能等。

数字视频编辑处理的另一个应用是虚拟演播室，它利用计算机生成运动或静止的三维场景，与现场拍摄的视频图像进行实时合成。在现场直播时，虚拟演播室系统综合生成的三维场景可以不断变化和更换，还能根据演播室摄像机运动的位置显示出正确的透视图，使合成的视频图像能取得真实的视觉效果。这样，虚拟演播室技术就使导演摆脱了时间、空间和道具制作等方面的约束，其创造力可以得

到充分的发挥。现在,虚拟演播室在电视、电影制作领域已经发挥着越来越大的作用。

目前有大量的数字视频编辑软件,根据其功能强大程度由小到大列举一些常见的编辑软件: move make, 会声会影, EDIUS, VEGAS, Premiere 等。其中以专业人员使用的功能复杂的 Premiere 为例,其主要功能包括: ①编辑和组接各种视频片段; ②对视频片段进行各种特技处理; ③在两段视频片段之间增加各种过渡效果; ④在视频片段之上叠加各种字母、图标和其他视频效果; ⑤给视像配音,并对音频片段进行编辑,调整音频与视频同步; ⑥改变视频特性参数如图像深度、视频帧率和音频采样等; ⑦设置音频、视像编码及压缩参数; ⑧输出生成 AVI 或 MOV 格式的数字音频文件,输出生成的文件可以在任何支持 Microsoft Video/QuickTime for Windows 格式的应用程序中播放; ⑨转换成 NTSC 或 PAL 的兼容色彩,以便换成模拟电视信号,通过录像机记录在磁带上或显示在电视上。由于 AVI 数据格式所采用的彩色系统与 NTSC 或 PAL 制式的模拟视频采用的色彩标准不同,因此,需要转换才能实现兼容; ⑩其他一些高级视频编辑功能。

参考文献

刘远东,徐兴洲,朱晓春. 视频编辑和制作. 北京: 人民邮电出版社,2005

(孙立峰)

shuzi shipin huoqu

数字视频获取(acquisition of digital video)

参见**图像获取**。

(孙立峰)

shuzi shuju wang

数字数据网(digital data network, DDN)

基于时分复用技术,利用数字信道提供半永久性连接,实现话音、数据、图像、实时信息等数据传输的数字通信网络。DDN 提供的电路速率通常在 2.4 ~ 2048Kb/s 之间。

如图 1 所示,DDN 由数字传输链路、数字交叉复用设备(DDN 节点)、网管系统和用户环路组成。其中,数字传输链路基于光纤、数字微波或卫星;数字交叉连接复用设备实现对数字电路的半永久交叉连接和子速率的复用;网管系统进行网络结构和业务的配置,实时地监视网络运行情况,收集和统计网络信息等。数据终端设备(data terminal equipment, DTE)通过数据服务单元(data service unit, DSU)与就近的 DDN 节点相连,其中 DTE 表示接入 DDN 的用户端设备,可以是普通计算机或局域网服务器、路由器,也可以是一般的传真机、电传机、电话机等; DSU 可以是调制解调器、基带传输设备以及时分复用、语音/数字复用等设备。

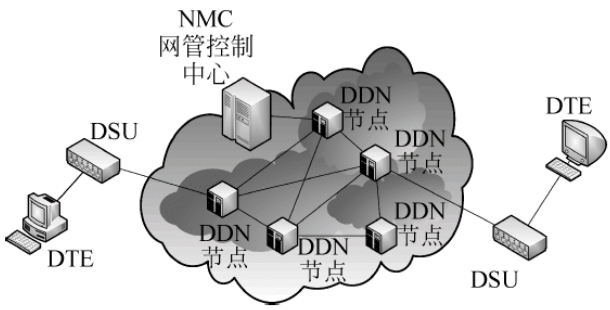


图 1 DDN 网络结构示意图

DDN 可向用户提供 2.4 K、4.8 K、9.6 K、19.2 K、$N\times64$ K($N=1\sim31$)及 2048 Kb/s 速率的点到点或点到多点的全透明数字专用电路,从而实现用户终端的接入、局域网/广域网的互连以及话音、数据、图像等的传输。DDN 具有专线专用、质量稳定、安全可靠等特点,适用于对数据传输质量和实时性、保密性要求高的业务,如政务、商业、金融、电子商务等领域的应用。

中国公用数字数据网(CHINA DDN)于 1994 年正式开通,覆盖到了全国所有省会城市、绝大部分地市和部分经济发达的县城。CHINA DDN 包括国家级 DDN、省级 DDN 和地市级 DDN。虽然 DDN 作为高质量数据专线的重要提供手段,曾得到了广泛的应用,并且目前在某些地区和领域还在继续发挥作用,但随着 IP 技术的兴起,数据通信网在向更高速率和分组化的方向发展,基于时分复用的低速数据通信业务 DDN 面临逐步萎缩并最终退网的命运。

参考文献

陈炽文, 张建红. 数字数据网. 北京: 清华大学出版社, 1999

(唐雄燕)

shuzi tushuguan

数字图书馆(digital library) 采用信息技术将各种载体上的文献信息以数字化形式进行组织、存储和管理,并通过**计算机网络**和各种通信手段为用户提供有效服务的信息集合或信息仓库。有人也把数字图书馆称为网络环境下的文献信息数据库系统或数据中心。从应用的观点看,数字图书馆是一

种数字化的图书馆**应用软件系统**。数字图书馆提供的基本服务主要有浏览、检索、个性化推荐、文献传递、电子参考咨询等,可以广泛应用于图书馆、文化教育、**电子商务系统**、**电子政务系统**、**物联网**以及智慧城市等领域,其最终目标是实现任何人在任何时候任何地点均能通过网络访问人类所有已记载知识的梦想。

20 世纪 90 年代以来,随着**数据通信**、**计算机网络**、存储技术、**多媒体技术**和**计算机软件**的发展,以及用户信息需求的不断增加,世界各国均投入大量人力、物力和财力进行数字图书馆建设。1993 年由美国国家科学基金会(NSF)、美国国防高级研究计划署(DAPAR)和国家航空与太空总署(NASA)联合提出了“数字图书馆先导计划”(DLI)。1994 年 9 月,一期工程(DLI-1)开始实施。其后,各国启动了多个数字图书馆计划或工程,主要有:美国国家数字图书馆,美国国家科学数字图书馆,加州大学数字图书馆,“G8”数字图书馆联盟,法国数字图书馆,欧盟数字图书馆,英国电子图书馆,澳大利亚数字图书馆,日本关西数字图书馆,Google Books 数字图书馆等。1997 年,“中国试验型数字图书馆项目”正式立项,标志着我国数字图书馆建设拉开序幕,其后的主要工程与计划有:国家数字图书馆,大学数字图书馆国际合作计划,中国高等教育文献保障系统,中国科学院国家科学图书馆等。

围绕数字图书馆各类资源的采集、组织、存储、管理、交换与服务,采用的主要技术有:①标准规范 涵盖资源的数字化加工、组织、管理、互操作、安全和服务等方面,包括数字资源的加工规范、元数据规范、数字对象规范、数字资源的发布规范、仓储规范、服务协议规范与接口标准、数字图书馆安全规范和版权管理规范等。②资源的数字化制作 包括图书和多媒体资源的采集、转换、组织、压缩和整理,其中的图书数字化制作涉及图书扫描、图像处理与识别、图像格式转换、元数据制作、目录制作、电子书封装以及质量一致性检查等技术。③数字资源的深加工 涵盖数字资源的处理、加工和整合等方面,包括多媒体信息的标注、注解分类、关联、浓缩、清洗、提纯等技术。④海量非结构化数据的存储与管理 包括数字对象的组织、高可靠性和高效的分布式存储、融合信息抽取、特征提取、全文搜索、多媒体搜索、文件搜索、数据挖掘、海量数据的高效索引和分布式计算等技术。⑤跨媒体的知识组织与检索 包括海量跨媒体数据的知识表达与索引、多媒体内容单元分割、智能处理与关联分析、融合与集成、跨媒体搜索以及排序等技术。⑥数字图书馆服务。包括资源导航、元数据检索、全文检索、基于内容的多媒体检索、跨语言检索、个性化服务、文献传递等服务以及数字借阅、微型标注、移动服务等新型增值服务技术。⑦数字内容的安全与保护 包括数字版权管理、基于数字水印和数字指纹的内容保护、基于内容的过滤和数据的备份与恢复等技术以及解决资源的非法盗取、复制、长期存取、计算机病毒和“黑客”攻击等数字信息的其他安全技术。

目前,数字图书馆正朝着智慧图书馆的方向发展。信息形式已经并正在从单纯的图书数据库扩展到由书籍、绘画、书法、影视、图纸、照片、网页以及上述信息之间的关系与衍生物构成的数据海;服务内容从传统图书馆服务扩展到智能服务,通过存储、分析、索引多样异构的海量非结构化数据,提供创新设计服务、超文本服务、主动服务、跨语言服务、多媒体服务、深度搜索等智能服务,使得数字图书馆更加主动化、专业化和智能化。

参考文献

Reddy R, Wladawsky-Berger Irving, et al. Digital libraries: universal access to human knowledge—a report to the President. President's Information Technology Advisory Committee (PITAC), Panel on Digital Libraries. 2001

(庄越挺)

shuzi tuxiang chuli

数字图像处理(digital image processing) 通过计算机对图像进行增强、复原、分割、去除噪声、特征提取等处理的过程、理论、方法和技术,以及以之为研究对象的学科。

发展简史

视觉是人类最重要的信息感知手段,数字图像处理又是计算机视觉的基础,并且是通信、遥感、医疗、气象、军事等诸多应用领域的有效工具。20 世纪 20 年代,图像处理首次应用于改善伦敦和纽约之间海底电缆发送的图片的质量。到 50 年代,数字计算机应用于图像处理之后,数字图像处理开始起步。1964 年美国喷气推进实验室用计算机对“徘徊者七号”太空船发回的大批月球照片进行处理,收到明显的效果。60 年代末,数字图像处理具备了比较完整的体系,形成了一门新兴的学科。70 年代,数字图像处理技术得到迅猛的发展,理论和方法进一步完善,应用范围更加广泛。在这一时期,图像处理主

要和模式识别及图像理解系统的研究相联系，如文字识别、医学图像处理、遥感图像的处理等。70 年代后期到现在，各个应用领域对数字图像处理提出越来越高的要求，促进了这门学科向更高级的方向发展。特别是在景物理解和计算机视觉（即机器视觉）方面，图像处理已由二维处理发展到三维理解或解释。随着互联网及数字视频技术的发展和应用的普及，图像处理在多种媒体的通信、存储、检索中发挥了重要的作用，核心技术包括图像压缩、图像检索、图像编码等。近年来，图像处理和计算机图形学相互结合，形成了基于图像的计算机绘制（Image-based rendering）、计算机动画和计算机游戏等方面的技术。数字图像处理已从一个专门的研究领域变成了科学研究、人机界面、多媒体娱乐/教育和日常互联网使用中的一种普遍应用的工具。

基本内容

主要目的

一般来讲，对图像进行处理（或加工、分析）的主要目的有：

（1）提高图像的视感质量，如进行图像的亮度、彩色变换，增强、抑制某些成分、对图像进行几何变换等，以改善图像的视觉效果。

（2）提取图像中所包含的某些特征或特殊信息。这些被提取的特征或信息往往为计算机分析图像提供方便。提取特征或信息的过程可以看作**模式识别**或**计算机视觉**的预处理。提取的特征可以包括很多方面，如频域特征、灰度或颜色特征、边界特征、区域特征、纹理特征、形状特征、拓扑特征以及关系结构等。

（3）图像数据的压缩与编码，以便于图像的存储、传输和保密。

主要内容

不管是何种目的的图像处理，都需要由计算机、图像获取设备、图像输出设备以及相关的软件来完成图像的输入、加工和输出处理。数字图像处理研究的内容主要有：

（1）图像获取和图像表示　主要是把模拟图像信号转化为计算机所能接受的数字形式，以及把数字图像按用户所需要的形式显示出来。

（2）图像复原　当造成**图像退化**的原因已知时，复原技术可用来进行图像的校正（参见**图像复原**）。复原技术是基于模型和数据的图像恢复，其目的是消除退化的影响，从而产生等价于理想成像系统所获得的图像。

（3）图像增强　当无法知道与图像退化有关的定量信息时，可以使用图像增强技术较为主观地改善图像的视觉效果。

（4）图像分析　对图像中的不同对象进行分割、特征提取（参见**图像分割**）和表示，从而有利于计算机对图像进行分类、识别、理解或解释。

（5）图像重建　由图像的多个一维投影重建该图像，可看成是一种特殊的图像复原技术（参见**图像重建**）。

（6）图像压缩和编码　对图像进行编码的主要目的是为了压缩数据和组织数据，便于图像的存储、传输和保密，特别是在互联网环境下的各种应用。当前的一些编码方法对图像数据的安全和知识产权的保护也提供了越来越多的措施（参见**图像的压缩编码**）。

主要数学工具

数字图像处理的数学工具可分为三大类。第一类包括各种正交变换和图像滤波等方法，其共同点是将图像变换到其他域（如频域）中进行处理（如滤波）后，再变换到原来的空间（域）中；第二类方法是直接在空间域中处理图像，它包括各种统计方法、微分方法及其他数学方法；第三类是数学形态学运算，它不同于常用的频域和空域的方法，是建立在积分几何和随机集合论的基础上的运算。

应用领域

数字图像处理主要应用于如下领域：

（1）通信　包括图像传输、电视电话、电视会议。

（2）遥感　无论是航空遥感或卫星遥感，都需要使用图像处理技术加工处理并提取有用的信息。遥感图像处理可用于矿藏勘探和森林、水利、海洋、农业等资源的调查，自然灾害预测预报，环境污染监测，气象卫星云图处理以及用于军事目的的地面目标识别。

（3）医疗诊断　如通过 X 射线、超声、计算机断层摄影（CT）、核磁共振等进行成像，结合图像处理与分析技术，进行疾病的分析与诊断。

（4）工业生产　主要有产品质量检测，生产过程的自动控制，计算机辅助设计与制造等。

（5）机器人视觉　通过实时的图像处理，对三维景物进行理解和识别，主要用于军事侦察、危险环境作业、自动流水线上装配工件的识别和定位，以及邮政、家庭服务等。

（6）出版、广告及视频制作　包括彩色图片的

绘制、编辑、加工和分色处理，电视制作中的图像特技处理和图像合成技术等。

(7) 军事、公安、档案管理等　如军事目标的侦察、制导和警戒系统、自动火器的控制及反伪装，指纹、手迹、印章、人像等的处理和辨识，古迹和图片档案的修复和管理等。

参考文献

1. Castleman K R. 数字图像处理. 朱志刚，等译. 北京：电子工业出版社，2002

2. Gonzalez R C，Woods R E. 数字图像处理. 2版. 阮秋琦，阮宇智，等译. 北京：电子工业出版社，2004

3. 许录平. 数字图像处理. 北京：科学出版社，2007 （朱志刚）

shuzi xitong moni jishu

数字系统模拟技术（simulation technologies for digital system）　对给定的数字系统在计算机上建立模型来模拟其行为、功能或性能的技术。它具有预测和验证双重意义。模型的建立和输入主要包括模拟对象各组件的功能特性、互连关系或相互作用关系、外部激励信号等描述信息，而这些信息通常是由专门的**硬件描述语言**（如 VHDL 或 Verilog）来描述提供。

根据模拟对象和模拟层次的不同，模拟主要分为两类，即电路模拟和逻辑模拟，其他还有系统模拟、器件模拟等。以下介绍电路模拟和逻辑模拟。

电路模拟又称电路分析。主要用于小规模数字系统的性能分析。自 20 世纪 60 年代起，人们就致力于计算机辅助电路模拟技术的研究。1962 年出现了第一个通用电路分析程序。集成电路进展所提出的需求推动了电路分析技术的发展，陆续出现了许多实用的软件。其中最著名的是美国加州大学伯克利分校开发的 SPICE 程序。这类程序的基础是关于电路的电压、电流和电阻的矩阵方程求解问题，其特点是精度很高但运算时间很长。优秀的模拟算法能够给出精确的直流特性、温度、定时、频率、灵敏度和噪声分析，其中定时分析主要是指电路的时延、竞争和冒险分析，在数字系统规模很大时尤为重要。于是，就出现了专门用于定时分析的软件，典型的如美国的 MOTIS 程序，它是在模型处理（例如允许门和晶体管作为模拟的基本单元同时存在）和数值方法上作了改进，它的求解速度比 SPICE 几乎快两个数量级，但这是以降低精度（约 10%）为代价的。电路模拟程序和定时分析程序两者在精度和时间上各有优势，在实际使用时往往把两者结合起来。对那些影响整个网络特性的关键部分或临界通路部分就采用电路模拟程序进行局部的、详尽的分析，而对整个电路则采用定时分析程序来分析。

逻辑模拟主要目的有两个：一是检查逻辑设计的正确性；二是进行故障模拟，产生故障诊断的测试码。按照组成的基本逻辑单元的规模大小，逻辑模拟可分为寄存器传输级、功能块级、门级和开关级逻辑模拟。寄存器传输级的基本单元是寄存器、存储器、总线和算术逻辑运算单元等系统组件；功能块级的基本单元是锁存器、触发器和计数器等功能部件；门级的基本单元是非门、与门、或门等各种逻辑门；开关级则将逻辑门进一步展开成晶体管，每个晶体管相当于一个开关，同时考虑晶体管导通电阻和节点电容的影响。开关级模拟常常作为介于逻辑模拟和电路模拟之间的一种单独的模拟方式。

随着电路规模和复杂度的提高，人们希望有一种混合的模拟程序能把逻辑模拟和电路模拟结合起来，以适应对模拟精度和速度的不同要求。美国的 SPLICE 程序是这类混合模拟程序的典型代表，它允许晶体管、门甚至功能块同时作为模拟的基本单元。选用何种模拟形式取决于设计对模拟对象、精度和时间的综合考虑。

模拟技术的最新进展是针对高速集成电路进行的定时验证技术。这是一种动态时序分析技术，是传统逻辑模拟技术的发展。它通过分析线路设计的全部时序组合，提供最坏情况的时序信息，即描述出信号的不定值区域。模拟技术的进一步发展将是各种不同的模拟手段相互补充、不断完善的过程。

参考文献

1. 王志功，景为平，孙玲. 集成电路设计技术与工具. 南京：东南大学出版社，2007

2. 阮刚. 集成电路工艺和器件的计算机模拟. 上海：复旦大学出版社，2007 （董云耀）

shuzi xiangji

数字相机（digital camera）　采用数字方式存储所拍摄图像的照相机。又称**数码相机**。数字相机用一组与普通照相机相同的光学镜头来成像，然后用一种被称为固体图像传感器的光敏半导体器件来获取图像。这种固体图像传感器是一种光敏芯片，它不仅对光作出反应，把光信号转换成电信号，而且通过其中的模数转换器把信号转换成数字信号，按

一定的格式将其压缩保存在相机内部或外接的存储设备中。通过接口,所存储的图像可以传输到计算机中保存、显示、处理和打印。数字相机还可以直接连接到专用的打印机上输出照片。

数字相机一般都带有一个小巧的液晶屏来观察拍摄场景和观看拍摄的相片,部分数字相机还支持连续视频图像拍摄。

数字相机的主要技术指标如下:

(1) 分辨率　单位为像素,常用的分辨率有640 像素 ×480 像素,800 像素 ×600 像素,1600 像素 ×1200 像素,以及 2400 像素 ×2000 像素和 3264 像素 ×2448 像素等。随着技术的进步,分辨率将越来越高,图像质量也越来越好。

(2) 固体图像传感器类型　目前主要有电荷耦合器件(CCD)和互补金属-氧化物-半导体(CMOS)两种图像传感器。CCD 的性能好,但价格和功耗较高,用于高档数字相机。CMOS 的性能比 CCD 差,但价格便宜,用于普及型相机。图像传感器以英寸(in)为单位,常见的有 1/2 in、1/3 in 等,尺寸越大拍摄效果越好。由于图像传感器只对光强响应,所以在拍摄彩色照片时,先用分光系统将光线分成三原色,再由图像传感器接收转换,然后用数字方式合成还原。

(3) 存储媒体类型和容量　数字相机一般用闪存芯片作存储器,有 SmartMedia, CompactFlash 和 Memory Stick 等几种形式,存储容量从 1 GB 到 32 GB 不等,最大可达 128 GB。部分数字相机支持可擦写光盘或者微型硬盘。

(4) 图像存储格式　数字相机支持 JPEG 或 Tiff 格式。JPEG 格式的压缩比大,节省空间,图像质量不如 Tiff 格式好。Tiff 格式是无损压缩,但占用较多的空间。

(5) 输出接口　主要有通用串行总线(USB)接口和 IEEE 1394。USB 接口是数字相机的标准配置,用于和计算机相连;IEEE 1394 速度较快,可以连接非 PC 设备。

(6) 微调焦系统　为了拍摄极近距离的物体,有些数字相机配有对镜头进行微调焦的系统。

数字相机产生于 20 世纪 70 年代。美国由于军事上的需要,用卫星在空中对目标进行拍照,推动了数字相机技术的产生与发展。20 世纪 90 年代伴随着数字影像与计算机技术的飞速发展,数字相机开始进入民用市场,1994 年诞生了第一款消费型数字相机。随着固体图像传感器技术和半导体存储技术的发展,数字相机迅速普及,图像质量也越来越高。通过高质量的打印机,图像输出质量已可与普通照片相比。目前,数字相机已经成为主流的照相设备,并已成为一种新型的计算机图像输入设备。

参考文献

1. Erickson B, Romano F. 数字相机与数字摄影技术. 承健,张耀炽,幺立勋,等译. 北京:电子工业出版社,1999

2. 祖厚. 数字照相机. 上海:上海科学技术文献出版社,2002

(谢长生　黄浩)

shuzi xinhao chuliqi

数字信号处理器(digital signal processor, DSP)　适合数字信号处理的专用处理器。随着微电子技术的发展,数字信号处理器常以大规模集成电路或超大规模集成电路芯片的形式实现,称为 DSP 芯片。DSP 芯片可以高速地实现过去由软件实现的大部分数字信号处理算法[如相关、卷积、滤波、**快速傅里叶变换**(FFT)等]。数字信号处理器大体上可分为两类:一类是专门为实现某种数字信号处理算法而设计的,仅适用于某些专门领域,称为专用数字信号处理器;另一类是可编程的,可以实现各种数字信号处理算法,从而可满足多个领域的需求,称为通用数字信号处理器。实现某种数字信号处理算法的 DSP 芯片将算法固化在其内部结构之中,例如专门实现 FFT 算法的 FFT 处理器,专门处理相关运算的数字相关器。又如英国 INMOS 公司生产的 A100 是专门用于处理一维滤波的 DSP 芯片,适合处理雷达、声呐和卫星通信系统的高速数据流;该公司的另一产品 A110 可高速完成一维和二维卷积运算,适合二维图像处理。这类数字信号处理器本身不需要编程,就能高效地实现数字信号处理算法。在实际应用时,将多个 DSP 芯片和微处理器(或计算机)以及其他器件或设备连接成一个系统,其中,DSP 芯片可高效实现数字信号处理算法,而微处理器则起控制和管理作用。可编程的数字信号处理器的体系结构有一系列特点。由于数字信号处理的主要算法都包含大量的乘法和加法,DSP 芯片内的乘法器和加法器可在 1 个时钟周期内完成 1 次乘法和 1 次累加,这种乘加运算还可不间断地连续执行。数字信号处理器采用哈佛结构,即程序和数据分别存放在独立的存储器中,并有独立的程序总线和数据总线,这允许取指令和执行指令在时间上完全重叠。处理器的指令系统除了包含程序存储器和数据

存储器之间的数据传送指令外，还能满足各种数字信号处理算法的特殊要求，例如相关运算中的连续乘加、FFT 运算中的倒序等。为了满足连续乘加等指令对数据量的需求，DSP 芯片中有多个地址生成器，可在 1 个时钟周期内生成多个存储器地址。另外，DSP 芯片还可将程序的循环机制用硬件实现，以减少开销。总之，数字信号处理器使多种操作并行执行，使数字信号处理算法得以高效实现。可编程的数字信号处理器在 20 世纪 80 年代初期已经得到了广泛使用。后来在芯片中增加了浮点乘加器，提高了运算精度和运算速度。90 年代以来，DSP 芯片都能提供多个高速通信口，可以方便地将多个这样的芯片组成多处理机并行处理系统，例如美国 TI 公司于 1994 年推出的 TMS 320C40 有 6 个 DMA 通信口，可构成三维的阵列处理系统。数字信号处理器已广泛用于图形与图像处理、语音信号处理、声呐与雷达信号处理、地震资料处理、数字通信、智能机器人等。

进入 21 世纪的 10 多年来，信号处理器不断采用新的内核结构，并得益于半导体技术的发展，其性能上升、成本和功耗降低，已成为通信、计算机、消费类电子产品等广大领域的基础器件，并在多个领域发挥了重要的作用。

当前在信号处理器制造领域，美国 TI(Texas Instruments)公司一直处于领先地位。占据了大部分国际市场。TI 公司的三种高性价比的信号处理器系列(TMS320C2000、TMS320C5000 和 TMS320C6000)在我国信号处理硬件领域应用广泛，TMS320C2000 系列主要用于工业控制领域，提供了很强的控制功能；TMS320C5000 系列为业界高性能的低功耗定点处理器，非常适合移动和手持设备的应用；TMS320C6000 主要用于高性能计算，最新的 TMS320C6455，时钟频率已达 1.2 GHz，单片处理能力可达到 9600MIPS。美国 Analog Device 公司也推出了 SHARC 系列和 Blackfin 系列的信号处理器，SHARC 系列常用于雷达、声呐领域，而 Blackfin 系列性能较高而功耗较低，常用于移动通信产品中。当前信号处理器正向着高性能内核、提高集成度、加强融合、低功耗和拓展多种应用的趋势发展。通过进一步完善内核结构，采用多通道和单指令多数据、超长指令字、超标量结构、多重流水线等技术，提高了处理器的性能。例如 TI 公司的 TMS320C64 + 系列处理器在 TMS320C64 内核的基础上增加了一些复杂计算的指令来改善内核结构，其性能提高了 30% 左右。提高集成度也是一个趋势，信号处理器厂商将多个信号处理器内核、大容量存储器和多种接口单元集成在一个芯片上，成为系统级电路。TI 公司在 2011 年上半年宣布推出最新的数字信号处理器 TMS320C6678，具有 8 个信号处理内核，能够提供 40GFLOPS 的浮点计算能力，其计算能力已相当于上一代产品和其他厂商的单核 DSP 的几十倍，该芯片还集成了 RapidIO 总线接口、PCIe 总线接口和千兆网络接口，单片即可构成强大的系统。

信号处理器与**微处理器**之间的融合，使得许多需要同时具有智能控制和数字信号处理两种功能的应用简化了设计，降低了成本。TI 公司的 DaVinci 数字视频处理器将微处理器 ARM9 内核和信号处理内核 TMS320C64 + 融合，兼有强大的数字信号处理能力和通用处理能力，非常适合嵌入式数字视频应用领域。

随着微电子技术的发展，常以超大规模集成电路芯片的形式来实现系统芯片(SOC)，数字信号处理器是在构建 SOC 中大量使用的必备组件，厂家以硬核或软核方式提供嵌入式 DSP 产品，用户可以非常方便地将 DSP 集成到自己的 SOC 芯片中。系统芯片(SOC)的广泛使用，进一步促进了 DSP 的迅猛发展。（侯朝焕　钟玉琢）

shuzi yinpin bianji

数字音频编辑(editing digital audio)　采用专门的软件对数字音频信息进行各种编辑处理以达到使用要求的过程和技术。在制作多媒体文档时，人们越来越多地需要自己录制和编辑数字声音(如背景音乐、解说词等)，一般是在个人计算机(PC)或工作站(workstation)上使用音频编辑软件完成。音频编辑软件有多种，较为常用的有：Cool Edit Pro、Adobe Audition。这些软件能方便直观地对数字声音(wav 文件)进行各种编辑处理(图 1)，通常包括如下一些功能：

(1) 基本编辑操作，例如声音的剪辑(删除、移动或复制一段声音，插入空白等)，声音音量调节(提高或降低音量，淡入、淡出处理等)，声音的反转，声音的倒置，持续时间的压缩、拉伸，消除噪声，静音，声音的频谱分析等。

(2) 声音效果处理，包括混响、回声、延迟、频率均衡、和声效果、动态效果、升降调、颤音等。

(3) 格式转换，如将不同取样频率和量化精度

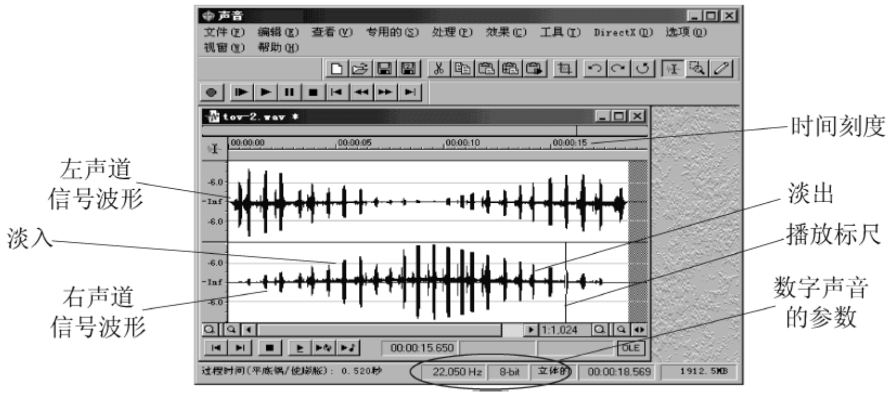


图 1　声音编辑软件的界面

的波形声音进行转换，将不同文件格式的波形声音相互转换，将存储波形音频文件（wav）格式声音与运动图像专家组第 3 层音频文件格式（MP3）声音相互转换，将 wav 音乐转换为乐器数字化音乐等。

（4）其他功能，如分轨录音，空间回旋，声道重混缩，为影视配音，刻录 CD 唱片等。

参考文献

张福炎，孙志挥．大学计算机信息技术教程．5 版．南京：南京大学出版社，2010

（张福炎　韩纪庆）

shuzi yinpin huoqu

数字音频获取（acquisition of digital audio）

使用特定的设备将声音信号经过数字化处理后输入计算机、手机等数字设备的过程。

声音是振动产生的波，是一种模拟信号。为了使用计算机等数字设备进行存储、处理和传输，首先必须将它转换成数字表示形式。声音信号数字化的过程如图 1 所示。取样的目的是为了把时间上连续的信号转换为时间上离散的一组样本。根据取样定理，取样频率不应低于声音频谱中最高信号频率的两倍。因此，语音信号的取样频率一般为 8 ~ 16 kHz，高保真音乐的取样频率应在 40 kHz 以上。量化实际上就是进行模数转换，即把每个样本用二进制整数来表示。声音信号的量化精度一般为 8 位、12 位或 16 位，量化精度越高，声音的保真度就越好。经过取样和量化之后，通常还必须进行数据压缩，以减少数据量，并按某种格式将数据进行组织，以便于计算机等数字设备存储和处理，或者在网络上进行传输。

将模拟声音信号转换成数字音频信号并使用数字设备进行存储、传输和处理有许多优点。例如，以数字形式储存的声音回放和复制时没有失真；数字声音的可编辑性强，易于进行效果处理；数字声音能进行数据压缩，传输时抗干扰能力强；数字声音容易与其他媒体（如文本、图像）相互结合；它也为自动音频检索和语音识别等应用创造了条件。

数字音频获取设备包括麦克风和声卡。麦克风的作用是将声波转换为电信号，然后由声卡进行数字化。声卡既参与声音的获取，也负责声音的重建，它控制并完成音频的输入与输出。主要功能包括：波形声音的获取与数字化；声音的重建与播放；乐器数字化接口（MIDI）声音的输入；MIDI 声音的合成等。

声卡的核心是数字信号处理器（DSP）。DSP 是一种专用的微处理器，它在完成数字声音的编码、解码、MIDI 声音合成及声音编辑操作中起着重要的作用。图 2 是声卡的原理框图，其中混音器的功能是将不同的声音信号进行混合，并进行功率放大和音量控制。

个人计算机中，随着主板技术的发展以及 CPU 性能的提高，同时也为了降低整机的成本，大多数声卡都已经集成在主板上。通常所说的“声卡”指的多半就是这种集成声卡，只有少数专业用的高档声卡才做成独立的插卡形式。

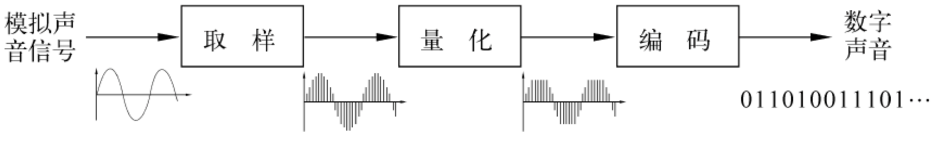


图 1　声音信号的数字化

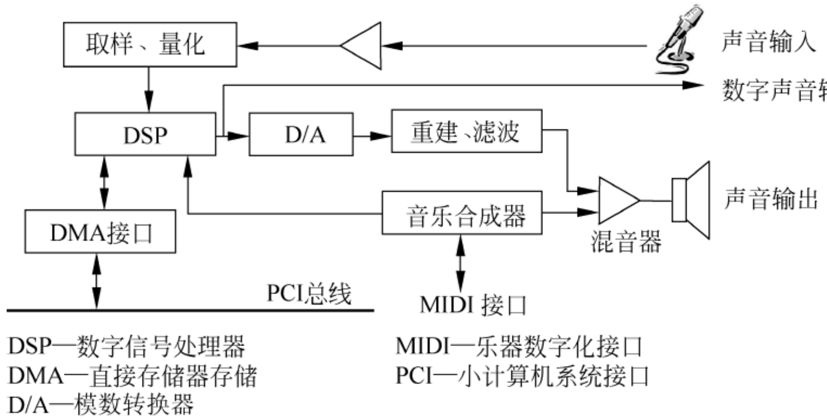


图 2　声卡的工作原理

除了利用声卡进行在线(on-line)声音获取之外,也可以使用数码录音笔进行离线(off-line)声音获取,然后再通过 USB 接口直接将已经数字化的声音数据从数码录音笔送入计算机中。数码录音笔的原理与上述过程基本相同,不过由于取样频率较低,仅适合录制语音使用。

参考文献

张福炎,孙志挥. 大学计算机信息技术教程. 5 版. 南京:南京大学出版社. 2010　(张福炎)

shuzi yinpin jiansuo

数字音频检索(digital audio retrieval)　通过音频特征分析,利用某种相似性测度查找出用户感兴趣的数字音频内容的技术。

纵观数字音频检索的发展历史,它是随着两类相关研究工作的不断深入而发展的。其一为传统的语音识别研究,其二为利用计算机对音乐等进行处理的研究。

20 世纪 90 年代,言语识别技术对广播新闻声音的处理进行了积极探索并有所突破,使人们能将大段的有声新闻自动转化为文本内容并进行保存。这样,使用成熟的文本检索技术就能进行言语文档的检索。同时,为有效地进行广播新闻识别,研究人员也开展了言语节目的切换、话音与其他乐音的分类、说话人转换检测等方面的研究。这些工作为大规模音频节目的索引构建提供了可能,促进了数字音频检索的发展。

20 世纪 70 年代,早期的工作是基于计算的音乐处理,开展了从声音中跟踪音乐结构的研究。这方面的工作为后来音乐检索中索引的构建等研究奠定了基础。

由于数字音频是连续的时间序列信号,因此需要先对连续的数字音频流进行分割,将其分为长短不一的音频单元;再对分割后的音频单元分析,并在此基础上进行检索。考虑到数字音频信号是典型的非平稳信号,为便于分析,一般都假设在一小段连续数字音频中信号是平稳的,然后对该小段内的数据进行特征分析,这样一小段的分析单元称为数字音频中的一帧。

数字音频检索可分为表示级检索(expression level retrieval)和语义级检索(semantic level retrieval)两大类。在表示级检索中,查询请求往往是对检索目标形式上的描述,或采用与目标音频形式相同的若干例子,并在声学特征层上从检索源中搜索与查询内容相同或类别相近的数字音频数据。在语义级检索中,通常是先从数字音频形式的查询请求中提取语义内容,或查询请求本身就是语义内容的描述,然后在语义特征层上从检索源中搜索与查询请求语义相似的音频数据。

数字音频检索的框架结构大体可用图 1 来描述。首先是索引的构建,通常是使用各种数字音频处理技术,先对原始声音数据进行数字化和特征提取,获得其在不同层次上的抽象信息,如使用语音识别技术获取音频数据的语义、使用音频分类技术得到不同音频数据的类别信息、结合自然语言处理技术归纳总结出音频篇章或段落的摘要等。利用上述这些信息可构造不同类型的索引库,通过索引库可快速检索到所需内容。检索时,根据用户的查询请求,通过检索模型利用索引库找到查询请求与数字音频库中相似的部分作为检索结果。

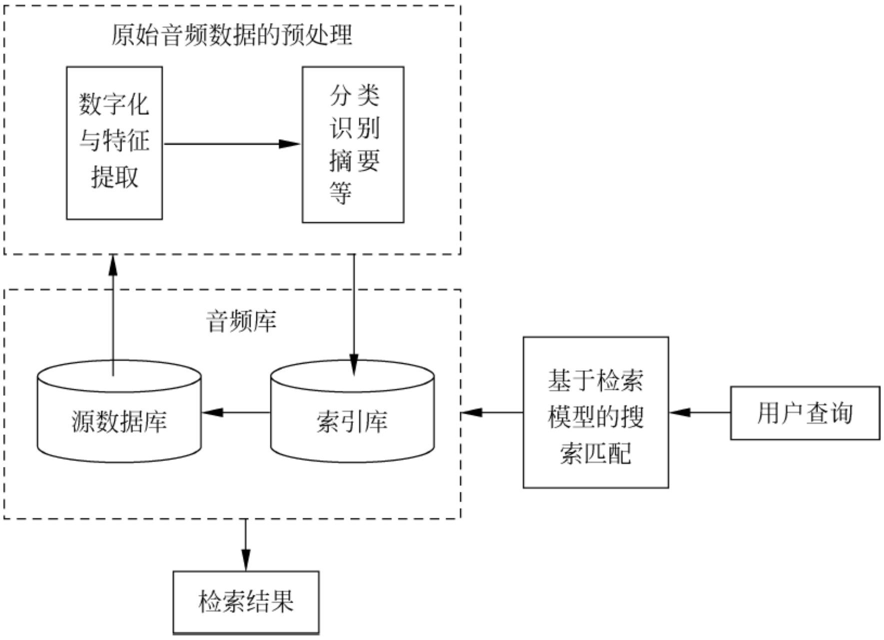


图 1 数字音频检索的框架结构

参考文献

韩纪庆,郑铁然,郑贵滨. 音频信息检索理论与技术. 北京:科学出版社,2011
（韩纪庆）

shuzi yonghu zhuanyongxian

数字用户专用线（digital subscriber line, DSL） 在现有的基于铜缆的提供电话服务的线路上传输高速数据的用户接入技术。主要有:***高比特率数字用户专用线***(HDSL)、***不对称数字用户专用线***(ADSL)、***甚高速数字用户专用线***(VDSL),统称为XDSL。在光纤接入网尚未普及时,用户迫切需要一种能够利用现有双绞线提供宽带业务接入的技术。XDSL接入技术可以在现有铜线用户专用线上经济有效地进行短距离数字通信,方便地实现宽带业务。

(1) 不对称数字用户专用线(ADSL) 是在无中继的用户环路上使用有负载电话线提供高速数字接入的技术。ADSL的速率是不对称的,其下行数字信道速率可达6 Mb/s,上行数字信道速率为144 Kb/s或384 Kb/s,可在现有任意双绞线上传输。它采用线路码,误码率低,用户的模拟话路独立。现在成熟的ADSL标准有两种,一个是G.992.1又称G.DMT,是ITU于1999年发布的全速率ADSL标准,上行传输速率为640 Kb/s,下行传输速率为6.144 Mb/s;另一个是G.992.2,又称G.Lite,是ITU于1999年发布的一种低速率ADSL标准,上行传输速率为512 Kb/s,下行传输速率为1.5 Mb/s。

(2) ADSL2 和 ADSL2 + G.992.3是对G.992.1的改进,这个增强的ADSL标准称为ADSL2,下行速率可达8 Mb/s,上行速率为800 Kb/s。ADSL2+是ITU于2005年发布的ADSL标准G.992.5,是ADSL2的增强,采用2.2 MHz的传输带宽,下行速率可达16 Mb/s,上行速率为800 Kb/s,ADSL2+在QoS保证、长线传输能力、噪声抑制、线路故障诊断等方面都比G.992.1有明显改进。

(3) 甚高速数字用户专用线(VDSL/或称VHDSL) ITU于2004年6月发布了VDSL标准G.993.1。VDSL是在ADSL基础上发展起来的,但采用了不同的编码方案,在很短的双绞线上可传送比ADSL更高速的数据。VDSL可在对称或不对称方式下运行。

(4) 第二代甚高比特率数字用户线路(VDSL2) 在ADSL2和第一代VDSL基础上发展的VDSL2标准G.993.2于2006年2月在ITU-T会议上获得正式通过,VDSL2标准将调制方式统一为DMT,其最高截止频率从12 MHz扩展到30 MHz,双向最大速率可达200 Mb/s。VDSL2通过扩展频谱、改善发射功率频谱密度等措施,可支持更高的传输速率和更远的传输距离,满足用户的高带宽需求,具有良好的发展前景,受到业界广泛关注。

参考文献

1. 胡道元. 智能建筑计算机网络工程. 北京:

清华大学出版社, 2002

2. 柯赓. 接入网技术与应用. 西安: 西安电子科技大学出版社, 2009 （程时端）

shuzu leixing

数组类型(array type) 分量类型均为同一类型的一种构造类型。实际问题中常遇到向量、矩阵之类数据结构,在**高级语言**中抽象成数组类型,一般写成如下形式:M:array D of R

这里 D 为数组 M 的下标类型,R 为数组 M 的分量类型。

从抽象的观点来看,一个数组可看成为由下标类型 D(定义域)到分量类型 R(值域)的映射。如果令 $i \in D$,则 $M[i]$ 表示对应 i 的数组分量。

数组类型的分量类型可以是某种**构造类型**,特别当其分量类型是数组类型时,这种数组称为多维数组。数据元素的数目由其下标类型的取值范围决定,如果数组的所有下标类型的取值范围可在编译时确定,那么这种数组称为**静态数组**,否则称为**动态数组**。从编译的角度考虑,静态数组可在编译时分配它的存储空间,而动态数组要到执行时刻才能分配它的存储空间。

字符串类型在有的高级语言中被定义为以字符类型为分量类型的一维数组,其下标类型用最小值为1,最大值大于1的子域类型定义。 （陈涵生）

shunxu chengxu sheji

顺序程序设计(sequential programming) 编写顺序程序的方法与过程。顺序程序的特点是其组成部分依次执行。其基本成分有顺序控制和数据控制两个方面。

顺序控制结构一般由语言文本定义了执行顺序。多数语言都将程序中语句的自然排列顺序定义为语句的执行顺序。程序人员可利用某些语句(如GOTO语句)改变语句的自然排列执行顺序。

顺序控制大致可分为三类:表达式内部(以及语句内部)的顺序控制,语句间的顺序控制,子程序(函数、过程)间的顺序控制。数据控制大多涉及命令式语言的基本特征——变量及其**绑定**。

变量是一个四元组,由变量名、变量属性、变量的值和变量的地址组成。由 W. Barron 给出的变量定义图,如图1所示。

变量名用来标识和引用变量。变量的重要属性

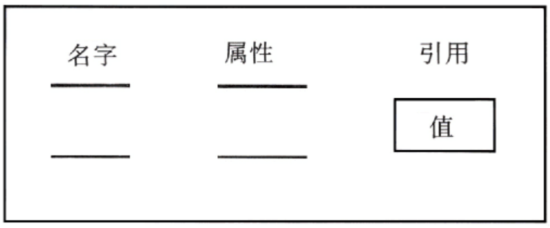


图1 变量的定义图

有其类型、作用域和生存期。变量的值和属性在运行过程中是可能改变的。将变量与值、属性联系起来,称为绑定。赋值语句中运算项的绑定时刻和表达式中的函数副作用,可能影响值的唯一性。

与子程序有关的数据控制问题主要表现在两方面:

(1) 进入子程序的数据控制问题在子程序体内可能出现三类名字:局部变量名、全局变量名和形式参数名。

过程调用的结果是通过变更其非局部环境完成计算,而函数调用的结果只是回送一个值,并不改变其非局部环境。函数副作用是由于函数执行过程中修改了其非局部环境或变量参数而引起的。

子程序调用时的实在参数求值及参数传递时的数据控制问题比较复杂。实在参数在求值之前必须先检查它与形式参数在个数、顺序、类型诸方面是否一致。参数传递有四种典型的方式:①按值调用;②引址调用;③按值-结果调用;④按名调用。与数据控制有关的还有参数的引用方式,共分三类:只读型、只写型和读写型。

(2) 从子程序回送的数据控制问题(结果回送)

共有三种方式:①若为函数,可将结果值存入函数名这一特殊变量后回送,或通过返回语句回送结果;②利用传址参数,或传值-结果参数回送结果;③通过非局部变量的副作用回送结果。

参考文献

1. 郭浩志. 程序设计语言概念. 长沙:国防科学技术大学出版社,1989

2. Pratt T W. Programming language: design and implememtation. Englewood Cliffs, NJ: Prentice-Hall Inc. ,1975 （郭浩志）

shunxu kongzhi

顺序控制(sequential control) 一种按照预先设定动作顺序,使生产设备、装置或生产过程按规定

时序,或在特定的输入条件作用下按预定的规律顺序动作的控制方式。顺序控制也称为程序控制。按照不同的使用要求,顺序控制主要有时序控制、条件控制和混合控制三种实现方法。时序控制是以时间为基准来控制每一步的动作,通常不需要反馈;条件控制是按照外部输入信号或者综合内部动作的结果作为下一步的动作条件,需要构成闭合的工作回路,完成顺序控制功能。混合控制就是既有时序控制,也有条件控制。所以,顺序控制系统可以是开环的、闭环的,也可以是开环和闭环混合使用的。

顺序控制的发展历史可以追溯到中世纪以前的各种顺序动作的机械。到了20世纪50年代,顺序控制才开始被广泛使用和发展起来。从顺序控制器的形式角度,顺序控制技术大致经历了机械式控制、电磁式控制、电子逻辑控制、可编程序控制等几个阶段。凸轮控制器是机械式顺序控制器的典型代表,随着轮轴的旋转,凸轮或凸轮组可以使与之相连的设备顺次工作。电磁式顺序控制器主要以普通继电器、延时继电器、时间继电器、接触器等器件实现有触点的逻辑控制关系,控制相关设备或过程的顺次动作。电子逻辑控制器是以晶体管或集成电路等半导体器件构成无触点的逻辑控制关系,进而控制相关设备或过程的顺次动作,这类控制器很好地解决了触点故障的问题,更适合于批量化的生产。可编程序式顺序控制器是一类以**微处理器**为核心、控制功能可以通过程序进行修改的顺序控制装置,其典型的应用包括基于**可编程控制器**的顺序控制器和基于**单片机**构成的顺序控制器,前者在数控、过程逻辑、联锁保护等控制方面有广泛的应用,后者则广泛应用在各种家用电器、自动导引、户外广告等单机式的应用场合。可编程序式顺序控制器无论在功能、可靠性、寿命、成本,还是在设计性能方面都具有优越性。所以,现代意义上的顺序控制器通常都是指可编程序式顺序控制器。

典型的顺序控制系统由顺序控制器、输入接口、输出接口及与计时、计数、通信、报警、指示等功能相关的接口组成(参见**可编程控制器**)。随着自动控制技术的不断发展,顺序控制的内涵也在不断拓宽,不仅要有强大的开关式逻辑控制功能,现在的顺序控制系统还包括了越来越多的模拟量控制功能。所以,顺序控制系统被广泛地应用于机械设备控制、自动生产线控制、运输和物流控制、家用电器和智能家居控制等工业或民用的自动化系统中。

参考文献

1. 白建云,杨晋萍. 火电厂顺序控制与热工保护. 北京:中国电力出版社,2009

2. 冈本裕生. 图解继电器与顺序控制器. 吕砚山,译. 北京:科学出版社,2008

3. 武居文雄,近藤贞雄. 顺序控制器入门. 丁尚瑾,寿端平,张志贤,等译. 北京:机械工业出版社,1981

(张光新)

shuohuaren shibie

说话人识别(speaker recognition) 计算机利用语音信号中所包含的反映说话人生理和行为特征的语音参数自动识别说话人身份的过程和技术。又称**声纹识别**。

对说话人(声纹)识别的研究可以追溯到20世纪40年代,1941年Bell实验室研制成功语谱图仪,L. G. Kersta通过观察语谱图进行识别,提出了“声纹”的概念。60年代,同是Bell实验室的S. Pruzansky提出了基于模式匹配和概率统计方差分析的说话人识别方法,出现了说话人自动识别研究的高潮。其间的工作主要集中在各种识别参数的提取、选择和实验上,并将倒谱和线性预测(LP)等方法应用于说话人识别。70年代末至今,说话人识别的研究重点转向对各种声学参数的线性或非线性处理以及新的模式匹配方法上,如动态时间规整(DTW)、主成分分析(PCA)、隐马尔可夫模型(HMM)、人工神经网络(ANN)以及多特征组合等技术。如今,说话人识别技术已逐渐进入实际应用,但仍存在许多问题需要解决,其中关键的特征表示问题(语音信号中哪些特征或其变换用来描述说话人是有效而可靠的)仍未能得到很好解决。

语音由发声器官运动产生。发声器官包括喉、声道、嘴和鼻。气流经过喉中的声门,引起声带周期性振动,形成周期性脉冲串进入声道,经过声道谐振以后,通过嘴或鼻孔向外辐射。不同说话人的发声器官具有不同的特性,例如声道的形状不同、嘴的开合度不一样等,另外,发声速度、韵律以及口音等也因说话人不同而存在差异。这些特定的说话人信息包含在语音中构成了说话人识别的物理基础。

说话人识别可以看作是言语(语音)识别的一种。但是,与言语识别不同的是,说话人识别希望从语音信号中提取出说话人的特征,而不考虑语音信号的语义内容。也即说话人识别强调说话人的个性,试图挖掘出包含在语音信号中的个性因素;而语

音识别则企求从不同人的语音信号中寻找共同因素,强调共性。

说话人识别的基本原理框图和生物特征识别几乎一致(参见**生物特征识别**),主要包括两大功能模块:特征提取和模式匹配。

特征提取是指从经过预处理的语音信号中提取出能唯一表现说话人身份的有效而稳定可靠的特征。目前说话人识别系统主要依靠语音的低层次声学特征进行识别,这些特征主要包括线性预测系数(LPC)及其派生参数、由语音频谱导出的参数(如基频 F0、共振峰 formant 等)、反映听觉特性的参数(如美尔频率倒谱系数 MFCC 等)以及上述参数的组合等。

模式匹配是将识别时的特征模板或模型与训练时得到的模板或模型作相似性匹配,计算它们之间的相似性(距离)。模式匹配的主要技术有动态时间规整(DTW)、向量量化(VQ)、隐马尔可夫模型(HMM)、**人工神经网络**(ANN)、高斯混合模型(GMM)、支持向量机(SVM)等,也可以将上述多种方法与不同特征进行有机组合来提高说话人识别的性能。

说话人识别可以分为两个范畴,说话人辨认和说话人确认。①说话人辨认是根据一段语音确定说话人是否已经注册,并判断其身份。②说话人确认是根据说话人的语音确定该说话人是否与他所声称的身份一致。

根据用户在使用系统时提供语音的发音材料,说话人识别可以分为与文本有关的、与文本无关的、文本提示的 3 种方式。①与文本有关的识别系统要求用户按照规定的内容发音,并根据特定的发音内容建立精确的模型,因而识别效果较好,但系统需要用户配合,如果用户的发音与规定的内容不符合,则无法正确识别该用户。②与文本无关的识别系统则不规定说话人的发音内容,因此建立精确的模型较为困难,识别效果较差。无论与文本有关还是无关,系统都无法区分是现场发音还是录音回放,文本提示的识别系统可以有效防止这种情况的发生。③文本提示的识别系统随机产生发音材料,并要求用户照此发音,因此仿冒者无法事先知道发音内容,但是该系统同样需要用户的配合。

说话人识别被认为是最自然的生物特征识别方式,因为语音的交互最易被人们接受。然而,若应用于大范围人群,语音并不能提供足够的信息来进行身份鉴别;同时,语音基本上属于一种行为特征,身体状况、紧张程度等因素会引起语音的改变。另外,环境噪声对语音识别的影响非常大。所以基于语音的身份鉴别系统的性能在很大程度上依赖于采集、传输、存储等数字化设备的精度,更重要的是该项技术需要说话人本身的配合。

从提高识别系统准确性和鲁棒性来说,说话人的语音识别与人脸识别、唇动识别等其他生物特征识别技术相结合(参见**多模态生物特征融合**)是行之有效的方法。

参考文献

1. Campbell J P. Speaker recognition: a tutorial. Proc IEEE, 1997, 85(9): 1437-1462

2. Kersta L G. Voiceprint identification. Nature Magazine, 1962, Dec. 29

3. Reynolds D A, Rose R C. Robust text-independent speaker identification using Gaussian mixture speaker models. IEEE Trans Speech and Audio Processing, 1995,3(1): 72-83

(蔡莲红　吴志勇)

shuoming

说明(declaration)　程序中用到实体的属性或定义描述。程序的说明部分使名字与被说明的实体相联系。例如,一个常量说明

PI:Constant: =3.1 415 926 536;

使 PI 与数学中的 π 值相联系。

说明可分为常量说明,变量说明,类型说明,子程序说明,模块说明,异常说明等。**变量说明**定义了程序中使用的各种变量及其属性。类型说明除了通常的**类型定义**外,还有私有类型说明。当希望说明一个能为外界使用,但又要隐蔽其具体实现和内部数据结构的类型时,可以使用私有类型说明。**子程序说明**参见**过程(函数)**。**模块说明**包括顺序模块说明与并发模块说明。**Ada 语言**中的程序包为一种典型的顺序模块,它描述具有相对独立性的一组逻辑上相关的实体。例如,定义复数类型的数据结构及其操作(+,-,*,/,求平方值)的程序包规约,在 Ada 中写为

```
package complex_arithmetic is
    type complex is
        record
            real-part:float:  =0,0;
            imag-part:float:  =0,0;
        end record;
    function "-"(a,b:complex)return complex;
    function "*"(a,b:complex)return complex;
```

```
    function "/"(a,b:complex)return complex;
    function sqr(a,b:complex)
end complex-arithmetic;
```

并发模块描述相对独立的可并行执行的一组逻辑实体。并发模块可在多机系统或多处理机系统上实现,也可在单处理机上以交叉执行的方式实现。例如,Ada中任务模块就是一种典型的并发模块,几个任务模块中各个任务除了发生会合的时刻外均为独立、并行地运行。异常说明是现代程序语言中,针对程序执行过程中可能发生的错误或异常情况作出处理的机制。这既是实时程序设计的需要,又可使程序的正常运行部分与出错处理(或异常处理)部分分离,提高了程序的清晰度与可理解性。

说明部分的主要任务是描述或定义各种实体,这里有一个作用域的概念,即每个说明的实体有这样的程序区域,在这个区域内可按定义给出的含义使用该名字。说明实体定义的作用域一般可静态确定,即由直接包含它的程序语法构造确定。与作用域相关的一个概念是作用范围,作用范围是指说明的标识符的可见范围。在分程序结构的语言中,内层作用范围可以不是外层作用范围的一部分。例如:以下Ada程序片段中,外部分程序A中内含一个内部分程序B,

```
    A: declare
alpha,beta:character;
begin
    get (alpha);
    B: declare
        alpha:integer;
        begin
            get (alpha);
            put (alpha);
        end B;
        put (alpha)
end A;
```

由于B中又说明了同名变量alpha,这样外部分程序A的alpha在B分程序中成为不可见的,因此,A的alpha的作用范围要去掉内部分程序B的这一区域。

程序语言的说明部分一般由编译程序在编译时处理完毕,但有的语言中(例如Ada)的一些说明工作需要在执行时才能处理。执行时处理说明部分的工作称为制作,它包括动态存储分配,计算初值表达式等工作。

参考文献

Ada Reference Mannal ANSI/MIL-STD-1875A-1983

(陈涵生)

sisuo

死锁(deadlock) 多个进程因竞争共享资源而处于永远等待的状态。例如,一个系统含有资源R_1,R_2,两个进程A,B分别占有其中一个资源而申请另一个资源,这样就出现了进程A占有资源R_1而等待进程B释放资源R_2;进程B占有资源R2而等待进程A释放资源R_1,于是这两个进程都不能执行下去而处于永远等待的死锁状态。

产生死锁的因素很多,不仅与系统拥有的资源数量有关,而且与资源分配策略,进程对资源的使用要求有关。出现死锁不仅会造成资源浪费,无法正常工作,严重时会使系统崩溃。因此,必须妥善解决和预防系统发生死锁。

有3种途径解决死锁问题:

(1) 死锁防止 是指采取措施保证系统不产生死锁。

系统产生死锁必须同时保持4个必要条件:①互斥条件 进程互斥使用资源;②部分分配条件 一个进程请求资源得不到满足而等待时,不释放已占有资源;③不抢占条件 任一进程不能从另一进程那里抢夺资源;④循环等待条件 存在一个循环等待链,其中,每个进程分别等待它前一个进程占有的资源。

只要能破坏上述4个必要条件的一个或几个,死锁就可防止。例如,系统采用资源静态分配法或资源层次分配法,则可破坏上述条件②或④,从而,可防止系统发生死锁。

(2) 死锁避免 是指采取措施避免系统产生死锁。

防止死锁发生代价太高,使资源利用率低,系统效率差。如果系统动态分配资源,在分配过程中,掌握每个进程的资源申请情况,当把资源分配给申请者会产生死锁的话,就拒绝分配;否则接受申请,为进程分配资源。这样可动态测出发生死锁的可能性并加以避免。银行家算法是著名的死锁避免算法。

(3) 死锁检测 是指采取措施找出系统中已发生的死锁并加以解决。

对资源的分配加以限制不利于进程对资源的共享和提高系统效率。解决死锁的另一条途径是死锁

检测,即对资源的分配不加任何限制,系统定时运行一个“死锁检测”程序,判断系统内是否发生死锁,若检测到死锁再设法加以解除。例如,可撤销某些有关进程,从一个进程那里剥夺资源等。进程资源图及 Petri 网等技术都可用来有效地判断系统是否发生死锁。

参考文献

孙钟秀,等. 操作系统教程. 4 版. 北京: 高等教育出版社,2008

(郑宇华)

SOUSUO

搜索(search) 在问题空间中,从初始状态出发,利用相关知识不断探求,找到某个目标状态的一种问题求解技术。

搜索是**人工智能**中的一个基本问题。因为人工智能研究的对象大多属于不良结构或非结构化的,一般难以获得其全部信息,也没有精确求解算法,只能依靠经验,利用已有知识逐步探索求解。另外,搜索也常用于求解那些具有较好的结构、理论上有求解算法但其复杂度高而难以在计算机上实施的问题,如组合爆炸问题。

人工智能中的搜索包括两个重要问题: 搜索什么,在哪里搜索。搜索什么是指搜索的目标,而在哪里搜索是指搜索空间。搜索空间是一系列状态的汇集,称为状态空间。人工智能中大多数问题的状态空间在问题求解之前不是全部知道的,所以搜索过程可以分成两个阶段: 状态空间的生成阶段和在该空间中对所求解问题目标状态的搜索阶段。一般地这两个阶段是交叉进行的。

根据搜索过程是否使用启发式信息,可以把搜索分为盲目搜索和**启发式搜索**。盲目搜索是一种系统的穷举搜索,它不考虑问题本身的特性,也不使用任何信息来指导搜索过程,而仅使用固定的策略产生状态和测试目标。它可用于许多不同的问题,但由于总是按照预定的路线进行搜索,只注重搜索过程而忽略搜索目标,因而具有很大的盲目性,效率较低,不利于复杂问题的求解。宽度优先搜索和深度优先搜索就是典型的盲目搜索。启发式搜索是在搜索中加入了与问题有关的启发性信息,用于指导搜索朝着最有希望的方向前进,从而加速问题的求解过程。启发式搜索在可接受的计算费用内去寻找最好的解,但不一定能保证所得解的最优性和可行性,甚至在多数情况下无法估计所得解同最优解的近似程度。**A^* 搜索**和**博弈树搜索**就属于启发式搜索。另外,随着研究的深入,自 20 世纪 80 年代以来,研究者基于客观世界中的一些自然现象,提出了一系列具有一定普适性的启发式搜索方法,称为**元启发式搜索**(也有人称为现代启发式搜索),主要包括禁忌搜索算法、模拟退火算法、遗传算法、蚁群优化算法、粒子群优化算法和人工神经网络算法等。

根据问题的表示方式,也可将搜索分为状态空间搜索和与/或树搜索。状态空间搜索是指用状态空间表示法来求解问题所进行的搜索。**状态空间表示法**是人工智能中最基本的形式化方法,是其他形式化方法和问题求解技术的出发点。所谓状态(state)就是表示问题求解过程中每一步问题状况的数据结构。操作(operator)也称为运算符,把问题从一种状态变换到另一种状态,即对一个问题状态使用某个可用的操作时,将使问题从一个状态变为另一个状态;操作可以是一个机械的步骤、过程、规则或算子,它指出了状态之间的关系。状态空间用来描述一个问题的全部状态及这些状态间的关系,通常用三元组$\langle S,F,G\rangle$表示,其中 S 为问题的所有初始状态的集合,F 为操作的集合,G 为目标状态的集合。与/或树搜索是指用问题归约方法来求解问题所进行的搜索,与状态空间法不同,它的基本思想是对问题进行分解或变换。即当问题比较复杂时,直接求解往往比较困难,此时可通过分解或变换将它转化为一系列较简单的问题,然后通过求解这些较简单的问题来实现对原问题的求解。

典型的盲目搜索策略有:

(1) 深度优先(depth-first) 是一种图搜索技术,它从根节点开始,沿着树的最大深度方向搜索,只有当上次访问的节点不是目标节点,且没有其他节点可以生成的时候,才转到其兄弟节点进行搜索。

(2) 宽度优先(breadth-first) 是一种图搜索技术,它从根节点开始,根据距根节点的距离,由近至远逐层搜索。

(3) 有界深度优先(depth-limited) 是深度优先搜索的一种改进,它限定了深度优先搜索的搜索深度。

(4) 迭代加深(iterative deepening) 是有界深度优先搜索的一种改进,初始搜索深度为 1,此后逐层增大。

典型的启发式搜索策略有:

(1) 最佳优先(best-first) 在搜索中,待计算的节点保存在 OPEN 表中,已计算过的节点保存在 CLOSED 表中。OPEN 表中的节点根据估价函数值

排序且优先搜索估价高的结点。

（2）爬山（hill-climbing）　是一种局部择优的方法，即在搜索过程中先扩展当前状态，然后选择其中最佳的状态进一步扩展。这种策略容易陷入局部最优。通常的解决办法是随机重启爬山过程。

（3）A^*　使用最佳优先策略的一种搜索方法。在这种策略中，设 OPEN 表的估价函数为 $f(n)=g(n)+h(n)$，其中 n 是搜索中遇到的任意状态，$g(n)$ 是从初始状态到 n 的代价，$h(n)$ 是对 n 到目标状态代价的启发式估计。如果 $h(n)$ 小于等于从 n 到目标的最短路径代价，则该算法就称为 **A^* 算法**。

（4）AO^*　与普通图的 A^* 算法相似，是与/或图的一种搜索策略。

（5）极小极大（min-max）　博弈树中的一种搜索策略，用在零和博弈中，即博弈方选择使自身优势最大化而令对手优势最小化的方案。

在搜索问题中，主要的工作是找到正确的搜索策略。一般地，搜索策略可以通过下面四个准则来评价：

（1）完备性　如果存在解，该策略是否保证能够找到它？

（2）时间复杂性　需要多长时间可以找到解？

（3）空间复杂性　执行搜索需要多大的存储空间？

（4）最优性　如果存在不同的几个解，该策略是否可以发现最高质量的解？

搜索策略反映了状态空间或问题空间扩展的方法，也决定了状态或问题的访问顺序。在盲目搜索中，搜索的顺序都是确定性的，即给定搜索空间后，其中各状态的遍历顺序也就给定了。对于启发式搜索来说，搜索的顺序是非确定性的，即在计算每个状态的参数之前无法确定下一步要扩展到哪个状态。在启发式搜索过程中，关键的一步就是如何选择下一个要考察的状态，不同的选择方法就形成不同的搜索策略。如果在选择状态时能够充分利用与问题有关的信息，估计出状态的重要性，就能在搜索过程中选择重要性较高的状态，从而有利于找到最优解。

目前，搜索的研究集中在启发式搜索和元启发式搜索上，特别是根据搜索问题的特点设计搜索能力更强的混合策略上。

参考文献

1. Luger G F. 人工智能：复杂问题求解的结构和策略（英文影印版·第 6 版）. 北京：机械工业出版社，2008

2. 贲可荣，张彦铎. 人工智能. 北京：清华大学出版社，2006

3. 邢文训，谢金星. 现代优化计算方法. 2 版. 北京：清华大学出版社，2005　（董兴业　黄厚宽）

sushu

素数（prime number）　除 1 和它自身外，不能被其他正整数整除的大于 1 的整数。例如 2，3，5，7，11，13，17 都是素数。除 1 以外不是素数的正整数称为复合数，1 是唯一的既非素数又非复合数的正整数。大约在公元前 300 年，欧几里得就证明了素数个数是无限的。目前所知道的最大素数是 $2^{74\,207\,281}-1$，共有 22 338 618 位，是 2016 年发现的。

称 $M_p=2^p-1$ 形式的数为梅森数，其中 p 是素数。当 M_p 是素数时，称为梅森素数。目前已知道 32 个梅森素数，最大素数就是最大的梅森素数 $M_{859\,433}$，最大的已知复合梅森数是 M_p，$p=39\,051\cdot2^{6001}-1$，是 W. Keller 于 1987 年发现的。梅森素数是否有无穷多个，这是数论中尚未解决的问题。

称 $F_n=2^{2^n}+1$ 形式的数为费马数，当 F_n 是素数时，叫做费马素数。目前只知道 5 个费马素数 3，5，17，257，65 537。目前所知的最大复合费马数是 $F_{23\,471}$，它有一个因子 $5\cdot2^{23\,473}+1$，是 W. Keller 于 1984 年发现的。是否有无穷多个费马素数，或者是否有无穷多个费马数是复合数，都是没有解决的问题。高斯曾经证明过，如果 F_n 是素数，那么可以用圆规与直尺作出正 F_n 边形，这说明费马数与几何中某些问题有深刻的内在联系。

两个差等于 2 的一对素数叫做孪生素数，例如 3 和 5，5 和 7，11 和 13 等，目前所知道的最大孪生素数是 $2\,996\,863\,034\,895\times2^{1\,290\,000}\pm1$，共有 388 342 位，于 2016 年发现。所谓孪生素数猜想是指存在无穷多对孪生素数。这个猜想至今没有解决，但认为它是正确的可能性很大。1966 年我国数学家陈景润得到的：存在无穷多个素数 p，使得 $p+2$ 是不超过两个素数的乘积。

关于素数个数的研究是素数论中最重要的问题之一。以 $\pi(x)$ 表示不大于 x 的素数个数，例如 $\pi(10)=4$，$\pi(100)=25$，$\pi(1000)=168$。前面说过的素数有无穷多个，即 $\lim\limits_{x\to\infty}\pi(x)=\infty$。此外容易证明 $\lim\limits_{x\to\infty}\dfrac{\pi(x)}{x}=0$，这说明素数出现的概率为零。所

谓素数定理是 $\lim_{x\to\infty}\frac{\pi(x)\ln x}{x}=1$，最早是勒让德和高斯提出的猜想，以后得到了证明。

如以 $\pi(x,q,l)$ 表示首项为 l，公差为 q 的算术级数中不大于 x 的素数个数，P. G. Lejeune Dirichlet 于 1837 年证明了：如果 l,q 是互素的正整数，那么 $\lim_{x\to\infty}\pi(x,q,l)=\infty$。1949 年 A. Selberg 得到了它的初等证明。类似于素数定理，对于固定的 q，容易证明 $\lim_{x\to\infty}\frac{\pi(x,q,l)\varphi(q)\ln x}{x}=1$，其中 $\varphi(q)$ 是欧拉函数，这就是算术级数中的素数定理。

参考文献

1. 华罗庚. 数论导引. 北京：科学出版社，1957

2. 华罗庚. 指数和的估计及其在数论中的应用. 北京：科学出版社，1963

（蒋增荣）

suxing ceshi

素性测试（primality test） 数论中确定整数 N 是素数还是复合数的方法。在许多情况，例如 RSA 公钥密码系统以及各种建立在有限域上离散对数问题的密码系统，都需要知道一个大整数是否是素数。近十几年来，这个问题发展极为迅速，目前已取得了突破，在 CDC CYBER 170-750 计算机上，大约 30 s 就可确定 100 位大整数的素性，约 8 min 可确定 200 位大整数的素性。最基本、最简单的素性测试是试除法，设 N 是奇整数，用一个奇整数 m 去除 N，若 m 整除 N，N 是复合数，否则，就称 N 通过了“m 试除”的素性测试。当 N 通过更多的试除测试时，N 越来越可能是素数，当 m 穷尽 3 到$\sqrt{N}$的一切奇整数时，N 就确实是素数了。试除法不仅能测试 N 的素性，还能分解出 N 的因子。但当 N 没有小因子时，试除法的运算时间太长，其时间复杂性是指数级的，在每秒百万次的计算机上，当 $N\approx10^{40}$时，算法时间差不多要 100 万年。

概率素性测试法是一种运算时间非常长（也许是无穷长）的测试法，但实际执行时，都有一个特殊的指令，当超过一定时间或一定步数后，算法就将停止，但是如果未达到规定的时间或步数算法便告结束，结果就是“真的”，即 N 确实是一个复合数。其一般形式是“对于输入奇整数 N，在 $[1,N-1]$ 中随机选取整数 b 作为基，对 N 关于 b 执行某种测试，一旦 N 在测试中失败，算法就停止。”当 N 是素数时，测试将无限进行下去，除非得到停止的指令。最有实际应用价值的是 Miller-Rabin 概率测试法，它进行如下的测试：设 N 为奇整数，记 $N-1=2^st$（t 为奇整数，$s>0$），若存在正整数 $b\in[1,N-1]$，满足 $b^t\equiv1\bmod N$，或 $b^{2^rt}\equiv-1\bmod N(0\leqslant r\leqslant s-1)$，则称 N 关于 b 通过测试，否则测试失败。由于任何素数都能通过测试，一旦对 N 测试失败，N 必为复合数。如果 N 通过测试，N 为复合数的概率至多是 $1/4$，通过 k 次，为复合数的概率不超过 $1/4^k$。k 越大，N 越可能是素数。Rabin 对 $(2^{400}-593)$ 进行了 $k=100$ 次测试均获通过后才断言它是一个素数，这个数是小于 2^{400}的最大素数。该方法的时间复杂度为 $O(\ln^3N)$。其他还有 Solovay-Strassen 概率测试法，它的时间复杂度也是 $O(\ln^3N)$。N 通过这种测试仍为复合数的概率不超过 $1/2$。确定型素性测试法是一种能证明 N 确实是素数的测试法。这类严格的素性证明必须完全排除其“伪”素数的可能，目前的算法大都或者只有理论意义或者非常复杂。例如根据 Wilson 定理，只要 N 整除 $(N-1)!+1$，N 就是素数，但它看来没有应用价值，因为当 N 是一个 3 位数时，$(N-1)!+1$ 就超过 100 位，计算量十分巨大。1876 年，E. Lucas 给出如下确定型测试法：如果同余式 $g^x\equiv1\bmod N$仅对于 $x=N-1$ 成立而对于 N 的其他真因子不成立，那么 N 是素数。为了求得 g，首先必须将 $N-1$ 进行分解，当 N 很大时，这是一件很困难的工作，目前最快方法的运算时间也是亚指数级的 $O(\exp(c\sqrt{\ln N\ln\ln N}))$。对于 $N-1$ 有小因子的整数 N，Lucas 测试法特别有效，特别可以得到有关费马数 $F_n=2^{2^n}+1$ 的 Pepin 测试法：当 F_n 整除 $(3^{(F_n-1)/2}+1)$时，F_n 是素数。同样，如能分解 $N+1$，也易于证明 N 的素性。Lucas 和 D. H. Lehmer 利用这点得到著名的有关梅森数 $M_p=2^p-1$ 的测试法：设 p 是素数，当且仅当 $v_{p-2}\equiv0\bmod M_p$ 时，M_p 是素数，其中 $v_0=0,v_s=v_{s-1}^2-2\ (s=1,2,\cdots,p-2)$。应用这个测试法和其他简单的测试法，Lucas 在 1976 年证明了 M_{127}是素数，这是使用计算机前找到的最大素数。1975 年，Lehmer 等人指出，只要知道 $N\pm1,N^2\pm N+1,N^2+1$ 的部分分解，就能证明 N 的素性，这些方法的最大成功是证明了 $2^{400}-593$ 是素数。1980 年一些学者利用“雅可比和”的概念提出了时间复杂度差不多是多项式 $O((\ln N)^{c\ln\ln\ln N})$的测试法。此外还有椭圆曲线测试法。目前已经证明，如果广义黎曼假设（ERH）成立，Miller-Rabin 概率测试法的时间复杂度是多项式 $O(\ln^5N)$的确定型

测试法,只要对 N 关于小于 $c\ln^2 N$ 的 b 进行 Miller 测试,如果均获通过,N 就是素数。

参考文献

1. Koblitz A. A course in number theory and cryptography. New York: Springer-Verlag,1987

2. Riesel H. Prime numbers and computer methods for factorization. Boston: Birkhäuser,1985

(蒋增荣)

suanfa

算法(algorithm) 解题过程的精确描述,由有限条可完全机械执行的、有确定结果的指令(或命令、语句)构成。关于问题求解,一方面是问题的描述,另一方面是用于求解此问题的装置。关于问题描述的一般要求是:题意准确、清晰、明了,解的规格确定。至于解题装置,可以是机器,也可以是人,也可以是两者的结合。算法是用计算装置能够理解的语言描述的解题过程,具有如下性质:

(1) 将它作用于所求解的问题的给定输入集上,或作用于问题自身的描述上,将产生唯一确定的动作序列,此序列是有限的;

(2) 此序列或终止于给出问题的解或终止于指出问题对此输入数据无解。

下面用一个例子来解释上述概念。问题描述是:"求实数 x 的平方根"。该问题的题意明确,但没有指明解的规格。如果满足于把"$\sqrt{x}$"看成是问题的解,那么,解决此问题的算法再简单不过了:"将根号 $\sqrt{\ }$ 与输入数据 x 相连接"。如果要求问题的解是精确的十进制表示,那么,2 的平方根就永远也算不出来。因为只用"有限"个动作是无法完成的。

如果把问题的描述改为:"求实数 x 的正平方根,准确到十进制表示的小数点后四位"。这个描述有许多改进:①它指明要求最终解是正根,而先前的那个描述对解的要求是不确定的;②它排除了把"$\sqrt{x}$"作为问题解的可能性;③"小数点后四位"提供了测试最终解的规格标准。

对于上述精确描述了的问题的解法可粗糙地表示为:

(1) 选择一个数 y,计算 y^2。

(2) 若 $|y^2-x|<5\times10^{-5}$,y 即是所要的解,否则返回到步骤(1)。

这个方法不是上述意义下的"算法"。因为选择 y 的初值的初始动作是不确定的,而且每个动作的后续动作(即如为何选择 y 的下一个值)也是不确定的,因此,将上述的过程作用于输入值 x 后无法得到"唯一确定的一串动作序列"。

若将上述的方法改进为:

(1) 选择 $y=1$。

(2) 计算 y^2。

(3) 若 $|y^2-x|<5\times10^{-5}$,y 即是所要的解,停止;否则,转到步骤(4)。

(4) 用 $((x/y)+y)/2$ 作为下步的 y 值;转到步骤(2)。

这个过程是牛顿–拉弗森方法的特例。它消除了上述粗糙方法的弊端。并且可以证明,将它作用于任何非负实数 x,在有限步之内一定会得到满足解的规格的正平方根。因此,这个过程已经非常接近于前述意义下的"算法"。只是,若将这个解法作用于负实数 x,解题过程就会无休止地计算一个又一个的 y,永不终结。即动作序列的有限性不能保证(这种不终结的过程称为半算法)。因此,需要对上述方法再作点改进,在它的前面加一步:

(0) 若 $x<0$,则无解,停止;否则,转到步骤(1)。这样就完美了,称得上是一个完全的"算法"。

上述算法是用自然语言描述的,易于为人所理解。不难用计算机所能接受的 PASCAL 语言将其改写成如下的程序:

```
program Square Root(input, output);
var X,Y,Z: real;
begin
    read ln(X);
    write ln ('Find the square root of', X: 0:
    0);
    if x≥0 then
      begin  Y: =1; Z: =Y*Y;
        while abs(X-Z)≥0.000 05  do
        begin Y: =((X/Y)+Y)/2;
              Z: =Y*Y
        end;
        write ln ('The square root of',X: 7:
              5,'is',Y: 7: 5,'·')
      end
        else
        write ln ('There is no square root for',
              X: 8: 5,'·')
end
```

这个程序执行时将输入所要求根的数据,然后给出回答:或给出平方根,或指出无解。这个程序对输

入数据 85,1,0.618 25, -9 的相继执行结果是:

Find the square root of 85

The square root of 85.000 00 is 9.219 54.

Find the square root of 1

The square root of 1.000 00 is 1.000 00.

Find the square root of 0.618 25

The square root of 0.618 25 is 0.786 28.

Find the square root of -9

There is no square root for -9.

在上述关于"算法"的严格定义中,我们坚持算法作用于给定的输入数据后所产生的动作序列的唯一性与有限性。唯一性蕴涵着确定性,有限性蕴涵着终止性。确定性指初始动作是确定的,每个动作的后续动作也是确定的("停止"动作无后续动作)。如果准许每个动作的后续动作可以有有限多种选择,即下一动作是不确定的,这种算法称为不确定算法。例如走迷宫,从其入口到出口,用不确定算法描述起来就比较简单。

设计算法与分析算法是计算机科学的核心问题。从应用范围来看,算法可分成数值算法和非数值算法两大类。从工作方式来看,算法可分成串行算法和并行算法两大类。早在 20 世纪 70 年代,D. E. Knuth 就指出,计算机科学就是研究算法的学问。因此,将算法这个概念严格化、精确化有重要的实际意义。

任何计算机程序至少是一个半算法,凡能到达"停止"的程序是一个算法(当然未必能解决程序人员心目中的问题)。对于一个给定的问题,可能有许多个算法(程序),但它们的质量不会全同。衡量程序质量的主要因素是,程序执行所费时间之长短和所需空间(存储容量)之多少。就当今的计算机硬件而言,尤以时间因素为贵。另一个重要的质量因素是算法的一般性问题(即适用范围之大小问题),如何裁剪一般性主要取决于应用背景。当然还有其他的质量因素,例如,对于数值算法而言,有收敛速度之高低,误差积累之快慢,结果精度之好坏,等等。关于算法的时、空效率参见**计算复杂性理论**。简言之,将算法概念严格化,既有利于算法设计又有利于算法质量分析。

把算法概念精确化还有深刻的理论意义。原先人们认为,任何合式描述的数学问题都是可计算的。20 世纪 20 年代开始,数学家对此提出了疑问:问题的"可解性"或函数的"可计算性"到底意味着什么?为了回答这个问题,几十年来建立了许多理论,如图灵机理论、递归函数理论、正规算法理论、λ 演算等等,并证明了所有这些理论是等价的:即,一个问题在一个概念下可解,在所有别的概念下也可解。这些研究启示了,算法过程是机械的,发明算法才是人类的创造。

参考文献

Aho A V, Hopcroft J E, Ullman J D. The design and analysis of computer algorithms. Boston, MA: Addison - Wesley, 1974

(朱洪 陈火旺)

suanfa sheji

算法设计(design of algorithm) 算法学的一个分支。将计算机应用中典型的求解需求抽象为计算问题,专门设计它们的求解算法,是算法设计的主要内容。一个计算问题需要明确描述它的输入数据和输出数据两个要素,输入数据包含的所有参量表示为问题的实例,输出数据的格式及应满足的条件表示为问题的询问。算法设计一般不能简单套用统一的规则,但在实践中,人们仍然归纳出一些具有普遍性的算法设计策略,如递归法、分治法、贪心法、动态规划、回溯搜索、局部搜索等。这些算法设计策略可指导人们设计新算法去解决新问题。算法的时间复杂性和空间复杂性,是刻画算法求解问题所消耗时间和占用空间的量化指标。在优化问题求解中,近似性能比是刻画算法求解的质量的指标。设计出算法,要分析算法的时间和空间复杂性,分析算法能够求得什么质量的解。根据算法的时间、空间复杂性、求解质量等性能指标,可比较算法的好坏,并考虑是否可设计出更好的算法,改进原来的性能指标。

参考文献

1. Aho A V, Hopcroft J E, Ullman J D. The design and analysis of computer algorithms. Boston, MA: Addison Wesley, 1974

2. Cormen T H, Leiserson C E, Rivest R L, et al. Introduction to algorithms. 3rd ed. Cambridge, MA: The MIT Press, 2009

3. 朱洪,陈增武,段振华,等. 算法设计和分析. 上海:上海科学技术出版社,1989

(朱洪 朱大铭)

suanfaxue

算法学(algorithmics) 系统地研究算法的设计、分析和验证的学科。算法是一个很古老的数学概念。最基本的算法是加、减、乘、除。它们是小学

生学习算术的入门方法。现代计算机出现以来，人们不断用计算机求解一些空前大型和复杂的数学问题。但是，凡与计算机打交道，无不是研究设计各种类型的算法（用某种计算机语言表示，如 PASCAL，并由计算机执行）。因此，将算法本身作为研究对象，建立一门独立的学科已是水到渠成。这就是算法学的基础。算法学并不包罗万象地研究设计各类数学问题的解的具体算法（这是各类具体计算学科的任务，如计算数学、计算力学、计算物理、计算化学等），而是从方法学角度，研究算法的设计、描述、确认和品质评审。算法学是计算机科学最重要的内容。有的计算机学者甚至称，计算机科学就是算法的科学。

算法学的主要内容大体可分为：设计、验证和分析。"设计"指创作算法的过程和研究有代表性的、好的创作策略（例如自顶向下策略，分而治之策略）。"验证"是指证明算法的正确性，即证明它满足所欲求解的问题的要求。最基本的证明手段之一是数学归纳法，通常用于证明循环不变式之成立。"分析"是指对算法的效率的确定。如算法所需执行时间之长短和运行空间之多少。时空效率通常表示为输入数据量的多少的函数。如果同一问题存在几个不同的算法解，分析的任务之一就是比较这些算法效率的高低。通常所说的算法复杂性就是指算法效率的高低。效率越低的算法意味着其复杂性越高；效率越高的算法意味着其复杂性越低。分析任务中最难的工作是研究问题复杂性（亦称计算复杂性），即确定一个问题的所有可能的算法解的最优效率或者最优效率的界限。

作为一个简单例子，考虑多项式的计算问题：$p(x)=a_n x^n+\cdots+a_1x+a_0$。

乍看起来，多项式计算不成问题，对任何 x，不难算出 $p(x)$ 的值。但若 n 非常大，或者我们面对着许多个 x 的值而必须计算许多个 $p(x)$ 的值时（这种情形可能来自编码理论或多项式逼近），这时就应该考虑设计更能胜任的计算方法了。

例如，按上述的多项式表示形式，采用最直接的算法，自左至右逐项孤立地计算，则计算 $a_n x^n$ 需做 n 次乘法，$a_{n-1}x^{n-1}$ 需做 $n-1$ 次乘法，如此等等，计算整个 $p(x)$ 需做 $n(n+1)/2$ 次乘法和 n 次加法。这样的算法，效率是很低的。试想一下，如果 x 的幂次不重复计算，那么乘法的次数将下降为 $2n-1$ 次。如果还坚持从左算到右，即从 $a_n x^n$ 算起，那就需要把 x 的幂次的中间结果保留下来，x^n，x^{n-1}，…，x^2 需要 $n-1$ 个单元。构造这一串中间值只需要 $n-1$ 次乘法。这样算法的效率为 $2n-1$ 次乘法、n 次加法和 $n-1$ 个中间结果单元。但若从多项式的右端算到左边，即从 a_0+a_1x 算起，那么 x 的幂次只需保留当前值即可，即仅需一个中间单元。这样算法的效率就更高一点，即为 $2n-1$ 次乘法，n 次加法和一个中间单元。这个算法可用类 PASCAL 语言表示如下：

基本算法

```
Poly ← a_0 + a_1 * x;
Power ← x;
for k = 2 to n do
Power ← Power * x;
Poly ← Poly + a_k * Power
end for;
output(Poly)
```

我们还可以把计算多项式的算法进一步优化，只需把 $p(x)$ 改写成如下的嵌套形式：

$$p(x)=(\cdots(a_n x+a_{n-1})x+\cdots)x+a_0$$

按照这种形式，采用嵌套乘、加算法，$p(x)$ 的计算仅需 n 次乘法和 n 次加法。这个算法可用类 PASCAL 语言表示为：

嵌套算法

```
Poly ← a_n;
for i = 1 to n do
Poly ← Poly * x + a_{n-i}
end for;
output(Poly)
```

这个算法的效率比前述的基本算法几乎要高出 1 倍。

嵌套算法有点不太直观，何以证明它是有效的，即它确实计算了多项式 $p(x)$ 呢？这可以通过检查循环不变式而得到证实。可用归纳法证明：在通过第 i 次循环之后，Poly $=\sum_{j=0}^{i} a_{n-j}x^{i-j}$ 成立（这个式子即所谓循环不变式）。当整个循环结束时，$i=n$，Poly $=\sum_{j=0}^{n} a_{n-j}x^{n-j}$，即 Poly $=p(x)$。

实际上，还可以证明嵌套算法是计算多项式的最优算法。嵌套计算方法是一种很老的算法，但证明它的效率最优性却是相当于后来的事情。

通过上述例子，讨论了计算多项式 $p(x)$ 的算法，验证了它们，并分析比较了它们的效率，最后指出嵌套算法的最优性。这就是算法学所关心的主要内容。多项式的计算虽然不难，但上述的展开过程

使我们看到研究算法学意义之所在。在算法学的领域中,难题比比皆是,许多新、老问题有待探索和研究解决。

参考文献

Baase S. 计算机算法:设计和分析引论. 朱洪,等译. 上海:复旦大学出版社,1985

(陈火旺　贲可荣)

suanshu luoji bujian

算术逻辑部件(arithmetic and logic unit, ALU)　对以机器字表示的操作数进行算术和逻辑运算的功能部件。它是数字计算机运算器的核心部件(参见**中央处理器**)。ALU 通常由**加法器**和功能控制电路组成。在有些计算机系统中,算术运算和逻辑运算分别用算术部件(AU)和逻辑部件实现。很多计算机不止一个 AU,而且 AU 常分成定点 AU 和浮点 AU,多处理机系统可包含多个相同的 ALU。

在数字计算机发展的历史中,美国 Intel 公司的 SN 74181 中规模集成电路是很流行的 4 位 ALU,该器件框图如图 1 所示。

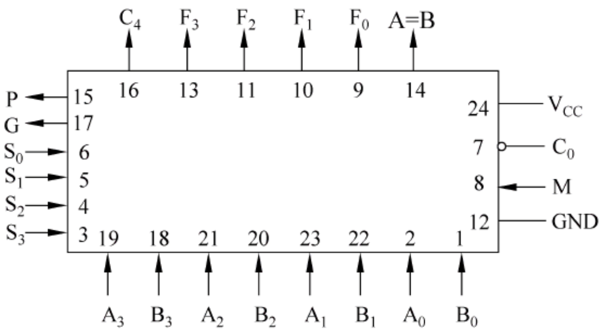


图 1　SN 74181 框图

图中的数字是引脚号,各引脚的功能如下:

(1) $A_0 \sim A_3$ 和 $B_0 \sim B_3$ 分别是两个 4 位数的输入。

(2) M 为控制位,高电平时,SN 74181 进行逻辑运算;低电平时,进行算术运算。

(3) $S_0 \sim S_3$ 是功能控制引脚,不同的组合 $S_3S_2S_1S_0$,可得到不同的功能,逻辑运算和算术运算各有 16 种。

(4) C_0 为低位进上来的进位信号。

(5) $F_3 \sim F_0$ 为算术逻辑运算的结果输出。

(6) C_4 为向高位的进位信号。

(7) 14 脚输出高电平,表示 A = B。

(8) P 和 G 是与并行进位链电路连接的输出信号,其中 G 为进位产生输出,P 称为进位传递输出。

SN 74181 的正逻辑功能表如表 1 所示。在功能表中,“加”表示算术加,“ + ”表示逻辑加;“减”表示算术减,“ - ”表示逻辑减。

表 1　SN 74181 正逻辑功能表

功能选择 S_3 S_2 S_1 S_0	M = H 逻辑运算	M = L 算术运算 $C_0 = 1$	M = L 算术运算 $C_0 = 0$
L L L L	$\overline{A}$	A	A 加 1
L L L H	$\overline{A+B}$	A + B	(A + B)加 1
L L H L	$\overline{A} \cdot B$	$A+\overline{B}$	$(A+\overline{B})$加 1
L L H H	“0”	减 1	“0”
L H L L	$\overline{A \cdot B}$	A 加$(A \cdot \overline{B})$	A 加$(A \cdot \overline{B})$加 1
L H L H	$\overline{B}$	$(A \cdot \overline{B})$加(A + B)	$(A \cdot \overline{B})$加(A + B)加 1
L H H L	$A \oplus B$	A 减 B 减 1	A 减 B
L H H H	$A \cdot \overline{B}$	$A \cdot \overline{B}$减 1	$A \cdot \overline{B}$
H L L L	$\overline{A}+B$	A 加(A · B)	A 加(A · B)加 1
H L L H	$\overline{A \oplus B}$	A 加 B	A 加 B 加 1
H L H L	B	(A · B)加$(A+\overline{B})$	(A · B)加$(A+\overline{B})$加 1
H L H H	A · B	(A · B)减 1	A · B
H H L L	“1”	A 加 A	A 加 A 加 1
H H L H	$A+\overline{B}$	A 加(A + B)	A 加(A + B)加 1
H H H L	A + B	A 加$(A+\overline{B})$	A 加$(A+\overline{B})$加 1
H H H H	A	A 减 1	A

SN 74181 是由非门、与或非门、与非门和异或门构成的组合逻辑电路,其基本逻辑结构是超前进位加法器。如果把 SN 74181 电路中的并行进位链省略掉,则其 1 位的逻辑电路图如图 2 所示。

从图 2 可以看出,三与或非门受 S_0 和 S_1 控制,二与或非门受 S_2 和 S_3 控制。S_0S_1 与 X_i 的关系及 S_2S_3 与 Y_i 的关系如表 2 所示。

当控制端 M 为高电平时,$F_i = \overline{X_i \oplus Y_i}$,根据 $S_3S_2S_1S_0$ 的不同取值,X_i 和 Y_i 就有 16 种组合,即得到表 1 的 16 种逻辑运算功能;当 M 为低电平时,$F_i = X_i \oplus Y_i \oplus \overline{C_i}$,也就是 $F_i = \overline{X_i \oplus Y_i \oplus C_i}$,即 F_i 为本位和的反码,这种情况下,X_i 和 Y_i 同样也有 16 种组合,实现 16 种算术运算功能。

采用 4 片 SN 74181,把它们的进位信号串接在一起,即可组成一个 16 位组间串行进位的 ALU。若用 4 片 SN 74181,再加上一片超前进位链(CLA)集成电路 SN 74182,即可组成一个 16 位组间并行进位的 ALU。

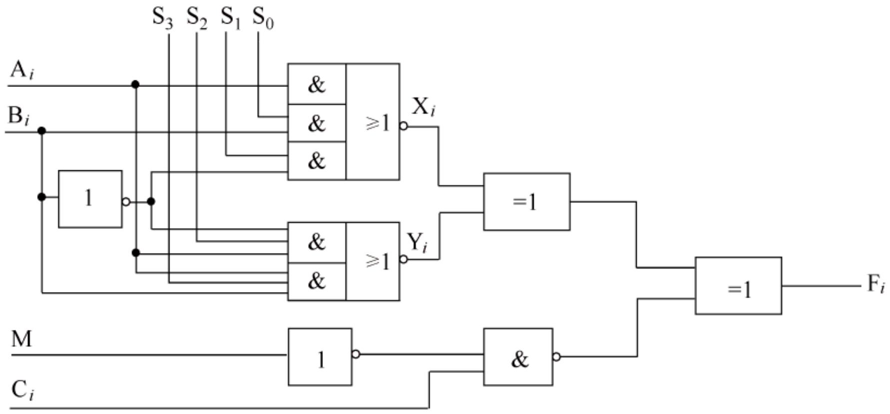


图 2 SN 74181 的 1 位逻辑电路图

表 2 X_i和 Y_i的值

S_1	S_0	X_i	S_3	S_2	Y_i
0	0	$\overline{A_i}$	0	0	1
0	1	$\overline{A_i}\cdot\overline{B_i}$	0	1	$\overline{A_i}+B_i$
1	0	$\overline{A_i}\cdot B_i$	1	0	$\overline{A_i}+\overline{B_i}$
1	1	0	1	1	$\overline{A_i}$

随着器件集成度的提高,可将更多位的 ALU 集成在一个芯片内。例如,20 世纪 80 年代后期 AMD 公司的 AM 29332 就是一个 32 位 ALU 器件。但是在**微处理器**出现以后,ALU 就不再作为独立的电路出现,而只是微处理器芯片的一部分。例如,8086 微处理器中,ALU 和运算寄存器、标志寄存器及**通用寄存器**一起构成 8086 的执行部件。

对于 80386,ALU 只是执行定点加、减类运算和逻辑运算的部件,乘除部件独立出来,而浮点运算则靠 80387 完成。到了 Pentium(奔腾)处理器,有两个整数 ALU,而浮点部件又由加法单元、乘法单元和除法单元组成。

尽管器件不同,但这些微处理器的 ALU 的基本原理是相似的。

参考文献

1. 李文兵. 计算机硬件原理教程. 天津:天津科技翻译出版公司,1994

2. 王爱英. 计算机组成与结构. 3 版. 北京:清华大学出版社,2001

(刘恩德)

suiji cunqu cunchuqi xinpian

随机存取存储器芯片(random access memory chip) 在给定的地址范围内,可对任意地址的存储单元进行写入和读出操作的**半导体存储器芯片**。随机存储器(RAM)是相对于只读存储器(ROM)和顺序访问存储器(如磁带)而言的,随机指的是访问时间与数据在存储器内的位置(地址)基本无关。存储在存储单元的数据在断电后可能会丢失。

根据数据存储方式,RAM 芯片主要分为静态 RAM(SRAM)芯片(参见**静态随机存取存储器芯片**)和动态 RAM(DRAM)芯片(参见**动态随机存取存储器芯片**)两种。双极型非同步 SRAM(通常称 SRAM 芯片)芯片是最早出现的 RAM 芯片,其数据存放在 R-S 触发器中。SRAM 具有简单的接口、快速的地址读出时间和读出写入周期,这些优点使其作为高速缓冲存储器的应用一直延续至今。但 SRAM 偏大的存储单元面积和功耗的缺点,使 SRAM 芯片的存储容量比不上后来出现的采用P/N沟金属氧化物-半导体(MOS)工艺技术非同步 DRAM 芯片(通常称 DRAM 芯片)。DRAM 芯片主要用于计算机的内存(主存),其数据位存储在与单管相连的电容中,在读出后数据会被破坏,需重新写入。长时间不读时数据也会因漏电而消失,必须周期性地对所存数据进行刷新(读出/重写),以防止数据消失。这就是称为动态 RAM 的原因。而 SRAM 芯片则不需要刷新。对这两种芯片,存储在存储单元的数据在断电后都会丢失。

随着 CPU 工作频率的不断提高,对高速缓存和内存速度和容量的要求也不断提升,SRAM 芯片和 DRAM 芯片的容量和速度也不断提升。除工艺技术的进步外,二者都先后增加时钟输入,以便同步地址、命令和数据接口,出现了同步 SRAM(SSRAM)和同步 DRAM(SDRAM)芯片,而内核仍是非同步的

SRAM 和 DRAM 电路。同步 RAM 芯片的主要优点是可实现突发(成组)数据的输入输出(即连续输入输出几个字)和流水线操作(即地址/命令的输入和数据的 I/O 可重叠进行),从而极大地提高了有效的数据传输速度。再进一步提高数据传输速率的技术是双倍数据速率(DDR),即在输入时钟的上升和下降边沿都可接收数据。这样,在突发传送时,数据的传送速率是芯片输入时钟频率的两倍。此外,四倍数据速率(QDR)SSRAM 将数据输入端口和数据输出端口分开,利用流水线操作交替输入读出命令及地址和写入命令及地址。读出数据和写入数据可分别以双倍数据速率从各自的端口输入和输出。这样,芯片的最大数据输出速率可达时钟频率的四倍。对 DDR/QDR 的同步 RAM 需特别关注数据、时钟和地址等高频信号传输的完整性。不仅要控制存储器芯片时钟输入的抖动、印制板线的长度、分叉和特性阻抗,而且要控制输出端口的输出阻抗和终端匹配,以保证在很窄的选通数据窗口内,能正确地接收和发送数据。

参考文献

1. Sharma A K. 先进半导体存储器——结构、设计与应用. 曾莹,等译. 北京:电子工业出版社,2005

2. 桑野雅彦. 存储器 IC 的应用技巧. 王庆,译. 北京:科学出版社,2006

3. http://www.samsung.com/Products/Semiconductor

(孙祖希)

suiji cunquji

随机存取机(random access machine, RAM) 分析算法复杂性的一种重要计算模型。现已证明,RAM 在计算能力上与图灵机等价。但在结构上,RAM 更接近通常计算机,因而,用它来分析算法的时空复杂度,结果更接近实际,也更具实用价值。

随机存取机的结构见图 1。它包括一条只读输入带,一条只写输出带,一个程序和相应的地址计数器,以及一个无限存储器。输入带分成小格,每格可放一整数。每读一个数,输入带头右移一格。输出带亦分成小格,每写一数,带头亦右移一格。RAM 的核心是程序。不同的程序代表不同的 RAM。程序是有机组成的一串指令。地址计数器标明要执行指令的地址。通常,每执行完一条指令,地址加 1。如遇转

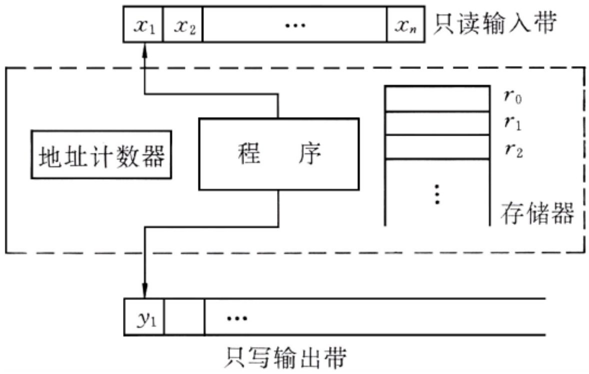


图 1　随机存取机

移指令,计数器内容作相应改变。存储器由无穷多个寄存器 $r_0, r_1, r_2, \cdots$ 组成。每个寄存器可放一整数。r_0 专作累加器,各种计算在此进行。

RAM 的指令系统包含通常计算机的各种基本算术逻辑指令。一个典型的指令系统包括 4 类 12 条指令:运算指令(加、减、乘、除),传输指令(读、写、存、取),转移指令(无条件转、零转、大于零转)和停机指令。指令可带标号,这有利于实现转移。操作数可用三种方式给定:立即数 $=i$,地址 i,间接地址 $*i$。在指令系统中增加一般指令,既不从本质上增加计算能力,也不在数量级上改变问题的复杂度。

RAM 可作为语言识别器,但更多时候是用作实现算法的计算机。一个算法的时空复杂度是执行相应 RAM 程序时实际耗费的时间和存储空间,它们与输入数据的长度有关。这里有两种核算方式:一致方式和对数方式。在一致方式下,每条指令占一个单位时间,每个寄存器占一个单位空间。在对数方式下,时空的核算比较精确,要考虑操作类型和操作数的位数。一般采用一致方式。

RAM 是一种串行计算模型。为分析并行算法复杂度,又提出并行随机存取机 P-RAM,例如互斥读写的 EREW P-RAM,并发读互斥写的 CREW P-RAM 和并发读写的 CRCW P-RAM 等。

参考文献

Aho A V, Hopcroft J E, Ullman J D. The design and analysis of computer algorithms. Reading, MA: Addison - Wesley, 1974

(徐美瑞)

suiji suanfa

随机算法(randomized algorithm) 一种使用了随机函数的算法,且随机函数的返回值直接或

者间接地影响算法的执行流程或执行结果。如设符号矩阵 $S[1..n]$, $n=2^{50}$, 满足 S 中元素只取值“M”或“F”,并且各为 $n/2$。则下面两算法是从 S 中找到取值为“M”的元素的随机算法:

算法 1　重复地以均一概率从 S 中取元素直到发现取出的元素的值为“M”为止;

算法 2　把算法 1 的重复次数限制到不超过 10 次。

由于两算法从 S 中取元素都是随机选择的,故称为随机算法。但不同的是,算法 1 一定可以找到满足要求的元素,但重复次数不确定;而算法 2 成功取得所需元素的概率为 $1-2^{-10}$,但该算法的时间复杂性为 $O(1)$。故此,人们又把随机算法分为两大类,即拉斯维加斯(Las Vegas)算法和蒙特卡罗(Monte Carlo)算法。拉斯维加斯算法的特点是,对计算问题,可以保证得到的计算结果是正确的;对组合优化问题可以保证其解是最佳的,但算法的时间复杂性函数是变化范围有界的随机变量,如上面的算法 1。蒙特卡洛算法的特点是,计算机时间复杂性是确定的,但只以概率 $1-f(n)$ 保证的计算结果是正确或最优。其中 $f(n)$ 是出错概率,如上面的算法 2。显然,若我们多次独立地执行时间复杂性为 $T(n)$ 的蒙特卡洛算法 $g(n)$ 次,并且每次执行的随机选择都是独立的,最后综合 $g(n)$ 次计算结果为最终的问题解,可以使得结果出错概率 $(f(n))^{g(n)}$ 小于某个常数 ε, $0<\varepsilon<1$。我们称蒙特卡洛算法为出错受限的。

相比传统的计算机算法,随机算法有以下特点:

(1) 算法描述往往非常简单,且容易实现。

(2) 某些拉斯维加斯算法往往是具有最佳平均时间复杂性的问题求解算法,并且满足算法时间复杂性和问题的输入无关,如在快速排序算法(quick-sort)中,对划分元素 y 的选取是随机地从数组中选取,该算法称为随机快速算法,期望时间复杂性为 $O(n\log n)$,其中,n 为数组中元素的个数,是期望时间复杂性最佳的排序算法。

(3) 对某些问题,有可能得到出错概率受限的蒙特卡洛算法,满足 $O(T(n)g(n))$ 比任何确定性算法的时间复杂性要低。其中 $T(n)$ 是算法时间复杂性,$g(n)$ 为独立执行的次数。如上面算法 2 的执行时间仅为 $O(1)$,而显然任何一个确定性算法的时间复杂性均为 $O(n)$。这种类型的随机算法无论是在理论上还是应用上都具有重要意义。

(4) 对有些问题的求解,目前只知道对该问题的随机算法,如对某些存在性问题的判断。对判断问题,按出错的类型上又可以把蒙特卡洛算法分为两类,即算法仅在回答是“对”(或者“错”)时,才发生错误,称为是单边错误;否则称为双边错误。显然对判断问题,拉斯维加斯算法是出错概率为零的蒙特卡洛算法。

早在百年前人们在科学研究中就采用了随机选择。随机算法目前在网络路由选择、集合运算、图论算法、布尔表达式的可满足性、矩阵和多项式验证以及密码学等领域,得到了比较广泛的应用。

参考文献

Motwani R, Raghavan P. Randomized algorithms. Cambridge, UK: Cambridge University Press, 1995

(马军)

suiji tu

随机图(random graph)　由随机过程产生的图。随机图的理论处于图论和概率论的交叉地带,主要研究各种典型的随机图的性质。一个随机图,可通过在一个顶点集合中的顶点之间,随机地连上边得到。随机图的随机性不仅仅反映在顶点,还反映在边。不同的随机图模型生成了边在图上的不同的概率分布。

如果将一含有 n 个顶点的集合记作 $V=\{1,2,\cdots,n\}$,将含有这 n 个顶点的所有的图的集合记作 G^n,那么两种最常见的随机图模型是 $G(n,M)$ 和 $G(n,P(\text{edge})=p)$。

$G(n,M)$ 表示顶点集合 $V=\{1,2,\cdots,n\}$ 上含有 M 条边的所有的图,且这些图都有相同的概率。因此,记 $N=\binom{n}{2}$, $0\leqslant M\leqslant N$,则 $G(n,M)$ 含有 $\binom{N}{M}$ 个元素,且这些元素有相同的发生概率 $\binom{N}{M}^{-1}$。M 可以看作 n 的一个函数,$M=M(n)$。一个图或图的集合 X 在 $G(n,M)$ 上发生的概率和期望可分别写作 $P_M(X)=P_{M(n)}(X)$ 和 $E_M(X)=E_{M(n)}(X)$。

$G(n,P(\text{edge})=p)$ 中 $0<p<1$,该模型表示顶点集合 $V=\{1,2,\cdots,n\}$ 上所有的图,且这些图上的边都是独立的以概率 p 连上的。也就是说,如果 G_0 是顶点集 V 上一个含有 m 条边的图,那么 G_0 发生的概率为 $P(\{G_0\})=P(G=G_0)=p^m q^{N-m}$, $q=1-p$。

$G(n,P(\text{edge})=p)$ 的一个自然改进模型是 $G(n,(p_{ij}))$,其中 $0\leqslant p_{ij}\leqslant 1$, $1\leqslant i<j\leqslant n$。该改进模型包含了顶点集合 $V=\{1,2,\cdots,n\}$ 上所有的图,这

些图上的边是随机独立连上的，且对于 $1 \leqslant i < j \leqslant n$，边 ij 的概率为 p_{ij}。$G(n,(p_{ij}))$ 的一个特殊情况是 $G(H;p)$，其中 H 是一个固定图且 $0 < p < 1$。在这种情况下，我们独立地以概率 p 为图 H 连上边。如果 $V(H) = V$，那么 $p_{ij} = \begin{cases} p, & ij \in E(H) \\ 0, & ij \notin E(H) \end{cases}$。

定义一个属性 Q(property) 为 G^n 的一个子集。那么，说"图 G 有属性 Q"就相当于"$G \in Q$"。如果任何时候都有若 $G \in Q$ 且 $G \subset H$，则 $H \in Q$；那么属性 Q 就称为单调增(monotone increase)或单调(monotone)。如果有若 $F \subset G \subset H$ 且 $F \in Q, H \in Q$，则 $G \in Q$；那么属性 Q 就称为凸属性(convex)。

最初，随机图的研究用来证明图的确定属性。后来，P. Erdös 和 A. Rényi 在"On the Evolution of Random Graphs"(1960)中提出：随机图的演化可以看作是通信网络的演化的简化模型。随着计算机技术的进步，随机图的概念和性质被认为可以用到现实复杂网络的研究中，比如社交网、互联网等。复杂网络有小世界(small-world)特征和无标度(scale-free)的特征，而传统的 Erdös-Rényi 随机图不具备这两个特征。2000 年以来，有研究者提出新的随机图模型用于复杂网络的研究。

参考文献

1. Bollobás B. Random graphs. Cambridge, UK: Cambridge University Press, 2001

2. Gross J L, Yellen J. Handbook of graph theory. CRC Press, 2003

3. Durrett R. Random graph dynamics. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge, UK: Cambridge University Press, 2007

(江志斌)

Sunzi dingli

孙子定理(Sun's theorem) 中国古代数学著作《孙子算经》中有"物不知其数"一问：今有物不知其数，三三数之剩二，五五数之剩三，七七数之剩二，问物几何？孙子对这个问题的解法可以表为："三人同行七十稀，五树梅花廿一枝，七子团圆正半月，除百零五便得知。"(见程大位《算法统宗》(1593 年))。它的意思为，以所求之数除以 3 的余数 2 乘 70，除以 5 的余数 3 乘 21，除以 7 的余数 2 乘 15，总加起来得 $2 \times 70 + 3 \times 21 + 2 \times 15 = 233$，减去两个 105，就得到答数 23。

"物不知其数"问题就是求最小正整数 x，使 $x \equiv 2 \pmod 3$，$x \equiv 3 \pmod 5$，$x \equiv 2 \pmod 7$。在解法中所用到的 70，21，15 三个数有如下性质：70 是用 3 除余 1，用 5 和 7 都除尽的数；21 是用 5 除余 1，用 3 和 7 都除尽的数；15 是用 7 除余 1，用 3 和 5 都除尽的数。因而 $2 \times 70 + 3 \times 21 + 2 \times 15$ 是用 3 除余 2，用 5 除余 3，用 7 除余 2 的数。而 105 是 3，5 和 7 的最小公倍数。

设 $m_i(i = 1,2,\cdots,k)$ 为正整数，$a_i(i = 1,2,\cdots,k)$ 为整数，联立一次同余方程 $x \equiv a_i \pmod{m_i}$ $(i = 1,2,\cdots,k)$，当且仅当 $a_i \equiv a_j \pmod{(m_i, m_j)}$ $(i \neq j)$ 时才有解，且解对于模 M(M 为 $m_1,\cdots,m_k$ 的最小公倍数)是唯一的。这就是孙子定理。上述解法也可推广到此一般情况，即先求出 $R_i(i = 1,\cdots,k)$，使 R_i 除以 m_i 的余数为 1，而能被 $m_j(j \neq i)$ 除尽。文献上也称此定理为中国剩余定理。这个定理及其求解的思想在数学中有广泛的应用。

参考文献

华罗庚. 数论导引. 北京：科学出版社，1957

(裴定一)

suoyin

索引(index) 一类特殊的数据结构，它由索引键和指向数据文件中相应记录的记录指针(或记录号)组成。索引键是记录的一个或一组数据项。索引用来提高数据查询效率，但同时引入存储代价和更新代价。索引结构如图 1 所示。

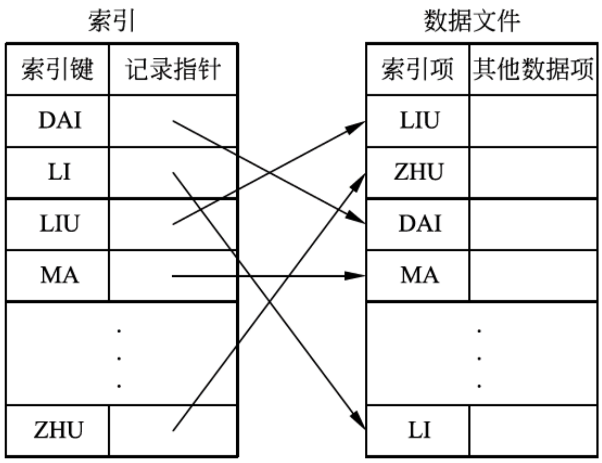


图 1 索引结构示意图

索引有不同的组织方式。评价这些不同类型索引的因素包括索引所占用的空间代价、索引支持的查询方式(等值查询、范围查询等)、利用索引获取记录的时间代价和索引更新的时间代价等。

索引的组织方式主要有线性索引、树形索引、散列索引和位图索引等。在线性索引中,索引项按索引键值排序,储存在顺序文件中。线性索引具有便于快速查找的优点和难于进行更新的缺点。树形索引是将索引组织成树形结构。**树形索引**既能进行快速查找,又易于索引结构的动态调整。最典型的树形索引是B树及其变种(B+树,B*树等)。散列索引将索引码和其地址指针组织为一个散列文件,散列函数将索引键映射到包含该索引码记录的数据块。通常使用动态散列技术来处理数据表动态变化。散列索引处理等值查询效率较高,但不支持范围查询。位图索引利用比特数组保存给定数据列的每一不同数值,支持将针对这一列的查询操作转换为比特数组的位运算。位图索引适合值域基数相对较小的数据列。

如果包含记录的数据文件按照某个索引键顺序排序,那么该索引键对应的索引为**聚集索引**。聚集索引也称为**主索引**。索引键指定的顺序和数据文件中记录的物理顺序不同的索引称为**非聚集索引**或**辅助索引**。如果在表上建立索引时指明为唯一性索引,则意味着在表中不允许有两个不同的记录在索引键上取值相同。

XML数据和图数据的查询处理也需要索引结构的支持。和关系数据上的索引不同,XML数据索引和图数据索引不仅仅包含节点数值的索引,而且包含节点间结构关系的索引。

参考文献

1. Garcia-Molina H, Ullman J D, Widom J. 数据库系统实现. 2版. 杨冬青,吴愈青,包小源,等译. 北京:机械工业出版社,2010

2. Silberschatz A, Korth Henry F, Sudarshan S. 数据库系统概念. 6版. 杨冬青,李红燕,唐世渭,等译. 北京:机械工业出版社,2006

(唐世渭 杨冬青)

T

taishi jisuanji

台式计算机(desk-top computer)　功能和结构上介于**工作站**和**笔记本计算机**之间的一种**个人计算机**。台式计算机与笔记本计算机的主要区别是，笔记本计算机的结构比台式计算机更紧凑、尺寸更小。台式计算机由于其有较小的体积，可以方便地放置于桌面上而得名。台式计算机从外表来看，通常由四部分组成，即机箱、**显示器**、**键盘**和**鼠标**。显示器、键盘和鼠标通常放置在桌面上；机箱在桌面上有两种放置方式，一种是机箱水平摆放在显示器下部；另一种是机箱垂直摆放在显示器旁。当然，现在机箱安放于桌子下面也是常见方式。机箱中包含主板、**中央处理器**、**主存储器**、**硬磁盘驱动器**、**软磁盘驱动器**、显卡、网卡和电源等。台式计算机常见于家庭和办公场所。

台式计算机的发展历史在2000年以前，基本等同于个人计算机的发展。尽管笔记本计算机很早就已经出现，但在2000年以前还只占个人计算机的很小部分。

台式计算机的构成可以参见个人计算机。

尽管未来台式计算机在个人计算机中比重逐步减小，但是它还将长期存在，尤其是在家庭中，大的平板显示器和较小的机箱会是一个趋势。另外就是将机箱内的部件全部装入平板显示器的后面，形成一种称为一体机的台式计算机。　　　　（侯紫峰）

tanxin fa

贪心法(greedy approach)　一种一般的试图求最佳解的算法设计方法。一个优化问题中的约束条件用于确定可能解，其中的最佳解使目标函数取极值。贪心法是一种多步决策法，它以某种最优化量度为根据，按该量度从大到小或从小到大的顺序来考察每一步。这种量度既可以是目标函数，又可以是其他函数。若采用目标函数，则每步优先考察使目标函数的值增长最快(用于求最大值问题)或增长最慢(用于求最小值问题)的可能解。

最优化量度的选择反映了“贪心”的角度，是采用贪心法能否求出真正最佳解的关键。对同一个问题，最优化量度可能存在多种选择，选得合适则可以求出真正最佳解，否则可能求出非最佳解，对后者往往是哪种选择好要视具体情况而定。一般来说，如果每步的“贪心”无法保证最终的全局最优，那么就不能保证采用贪心法设计的算法一定产生真正最佳解。总而言之，采用贪心法设计的算法本身不能保证求出的一定是真正最佳解，除非能够另行证明(如活动选择问题、实数背包问题、最佳合并顺序问题等)。证明的基本思路是证明算法产生的解不会比假设的真正最佳解差，证明过程中可能用到数学归纳法和分情况分析法等方法。因为需要按最优化量度从大到小或从小到大的顺序来考察每一步，所以排序是难免的，因此对采用贪心法设计的算法$O(n\log_2 n)$的执行时间是必需的，但相对而言其求解速度往往比较快。特别地，对许多NP完全的优化问题常用贪心法来设计求其近似最佳解的近似算法。

作为一种一般的算法设计方法，贪心法可用于求解实数背包问题、多处理机调度问题、带时限的作业调度问题、最佳合并顺序问题、磁盘文件的最佳存储问题、活动选择问题、任务调度问题、区间调度问题、最小延迟调度问题、最优高速缓存问题、图的最短路径问题、最小生成树问题、数据压缩编码问题、最小费用有向树问题、负载均衡问题、集合覆盖问题等优化问题。

参考文献

1. Aho A V, Hopcroft J E, Ullman J D. The design and analysis of computer algorithms. Boston, MA: Addison-Wesley, 1974

2. Cormen T H, Leiserson C E, Rivest R L, et al. Introduction to algorithms. 3rd ed. Cambridge, MA: The MIT Press, 2009

3. Kleinberg J, Tardos E. Algorithm design. Boston, MA: Addison-Wesley, 2005　　　　（殷建平）

taojiezi

套接字(socket)　在分布式软件系统中，一种用

于传输数据包并被赋予一个称为套接字号的整数作为唯一标识的进程间通信机制。套接字由应用程序通过操作系统创建。套接字的概念源自伯克利 Unix 操作系统 BSD(Berkeley Software Distribution)。

套接字通常包含网络(互联网)套接字、套接字地址和套接字编程接口三方面内容。网络套接字是本地套接字地址、远端套接字地址和传输协议三部分的唯一组合;套接字地址是包含本地 IP 地址及端口号的单一标识;套接字编程接口是应用程序开发进程间通信时涉及的编程规范。当操作系统收到一个 IP 包时,通过提取 IP 和传输协议头中的套接字地址和协议信息,将 IP 包中的有效数据传递给相关的应用**进程**或**线程**,从而实现包交换。套接字编程接口通常都遵循伯克利套接字(Berkeley sockets)标准,伯克利套接字是作为应用编程接口随 4.2BSD 操作系统一起发布的。将系统资源的操作映射为文件 I/O 的操作是 UNIX 的一种重要设计理念,套接字编程接口也属于一种文件 I/O 的**虚拟化**。

互联网套接字的类型主要有:①数据报套接字,也称无连接套接字,采用用户报协议 UDP,当同一个服务方给多个客户方发送同一数据包时,该服务方仅由一个服务方进程进行处理;②流套接字,也称面向链接的套接字,采用传输控制协议 TCP,当一个服务方给多个客户方发送同一数据包时,服务方需要一一对应地并发创建多个子进程,创建多个套接字连接;③原始套接字,主要用于路由器和其他网络设备。目前,在软件系统中使用的套接字大多是基于 TCP/IP 协议的。

套接字是网络传输层上的概念,在分布式软件的开发中属底层的通信机制。目前,广泛应用的各类**分布计算中间件**,其支持的通信机制实际上都是以套接字为基础实现的。

参考文献

史蒂文斯,芬纳,鲁道夫. UNIX 网络编程(卷1):套接字联网 API. 3 版. 北京:人民邮电出版社, 2010

(吴泉源　滕猛)

tezheng tiqu

特征提取(feature extraction)　当给定表征模式的 d 个特征时,通过变换用较少数量的 D 个特征有效地对模式进行识别的方法。

特征提取是用统计决策理论求解模式识别问题的重要组成部分,直接关系到**模式分类器**(参见**模式分类器**)的结构及性能。和特征选择(参见**特征选择**)一样,所求得的 D 个特征应具有很好的类别可分离性。当确定了可分离性准则后,特征提取过程就可以转化为求解一个优化问题。

如果把模式的初始特征写成向量形式,$\boldsymbol{y} = [y_1, y_2, \cdots, y_d]^{\mathrm{T}}$,那么通过线性变换 w,可以得到 D 个特征所组成的新的向量 $\boldsymbol{x} = [x_1, x_2, \cdots, x_D]^{\mathrm{T}}$,且 $\boldsymbol{x} = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{y}$,$\boldsymbol{w}$ 是 $d \times D$ 矩阵。通常的特征提取方法就是在给定的表征类别分离性的准则函数下,在所有的 $d \times D$ 矩阵中选取有最大准则函数值的 $\boldsymbol{w}$ 矩阵。这类方法被称为线性判别方法。

当数据的特征表示是稀疏的时候,可以在待优化的准则函数上加一个 L1 正则化项,通过解这个优化问题得到一组特征(稀疏解)来实现特征的提取。

基于 K-L 变换的特征提取方法在模式识别中得到了广泛的应用。它的基本出发点是通过 K-L 变换,消除了初始向量空间各分量之间的相关性,有可能去掉那些变换后对后续分析(如去噪,可视化,分类等)没有任何作用的坐标轴以达到降低特征空间维数的目的。

参考文献

1. Devijver P A, Kittler J. Pattern recognition: A statistical approach. Prentice-Hall, 1982

2. Fukunaga K. Introduction to statistical pattern recognition. 2nd ed. Boston: Academic Press, 1990

(边肇祺　张长水)

tezheng xuanze

特征选择(feature selection)　从表征模式的初始特征集合中选取一个子集的方法。

特征选择是用统计决策理论求解**模式识别**问题的重要组成部分,直接关系到**模式分类器**(参见**模式分类器**)的结构及性能。从初始 d 个特征中所求得的一个子集中的 D 个特征应具有很好的类别可分离性,即如果用 D 维空间中的一个点表示一个模式(特征向量),那么同类别的各个模式的特征向量应尽可能地集聚在一起,而属于不同类的各个模式的特征向量又要尽可能地互相远离。在数学上,我们可以定义一个表征上述要求的准则函数,使选出来的 D 个特征相对于其他的 D 个特征组合有最大的准则函数值。有以下几种直接从初始特征集合中求取有最大准则函数值的 D 个特征组合:

(1) 枚举搜索法　从 d 个特征中组合出所有的 D 个特征集合,分别求出其准则函数值,然后选出准

则函数值最大的 D 个特征。这是一种最优算法,其缺点是计算量大。因此一般只在特征数量很少时才采用。

(2) 次优算法　把单个准则函数值最大的 D 个特征组合在一起是一个最简单的次优算法,例如:Relief 方法。此外,也可以只用少量的特征进行组合计算,求出最好的特征组合后,再不断增加新的特征,直到得出所要求的 D 个特征。还可以从所有特征开始,不断去掉对准则函数值贡献最小的特征,直到减少到 D 个特征为止。理论分析表明,次优算法不能保证得到满意的结果。

(3) 一些智能优化算法也可以用于特征选择。例如:模拟退火方法,遗传算法,Tabu 搜索方法等。

参考文献

1. Devijver P A, Kittler J. Pattern recognition: A statistical approach. Prentice-Hall,1982

2. Fukunaga K. Introduction to statistical pattern recognition. 2nd ed. Boston: Academic Press,1990

(边肇祺　张长水)

tiaoma yueduqi

条码阅读器(bar code reader)　利用光电原理将条码信息转化为计算机可接受的信息的输入设备。它常用于图书馆、医院、书店以及超级市场,作为快速登记或结算的一种输入手段,对商品外包装上或印刷品上的条码信息直接阅读,并输入到联机系统中。

条码是由一组规则排列的条、空及其对应字符组成的标记,用以表示一定的信息。条是指条码中反射率较低的部分,空是指条码中反射率较高的部分。通常采用两种对比度高的颜色来分别构成条和空,例如,黑与白,蓝与黄,绿与红等。

由于条码中条和空排列规则的不同,形成各种类型的条码,如 UPC 条码、EAN 条码、二五条码、交错二五条码、三九条码、库德巴条码、中国标准书号(ISBN 部分)条码等。图 1 是按我国国家标准 GB 12904—1991 印制的一个通用商品条码,其结构与 EAN 条码相同,由 13 位数字码以及对应的条码组成。前 3 位数字为前缀码,标识国家或地区(我国为 690),后面 4 位为标识制造厂商的代码,再其后 5 位码是标识商品名称的代码,最后 1 位是校验码。条码阅读器通常带有一个发光装置,将光线照射到条码上,用光敏元件接收反射光。由于深浅不同的线条反射的光强度不同,得到高低不同的电平信号,经译码装置转换为一组数字信号。译码装置和光电传感装置可以做成一体,也可以是互相独立的两个部件。

图 1　条码

条码阅读器按工作方式可分为固定式和移动式(手持式),按光源的不同,可分为发光二极管、激光及其他光源形式。

最常见的条码阅读器是笔式阅读器,它属于手持式,俗称光笔,但与显示器屏幕光笔不是同一物品。它以发光二极管为光源,其操作如同人们用笔画一条线,只要将笔头的小窗口对准条码,在其表面匀速移动,条码信号就通过电缆进入计算机。其结构原理如图 2 所示。这种阅读器使用方便,价格低廉,但必须将它与条码接触。

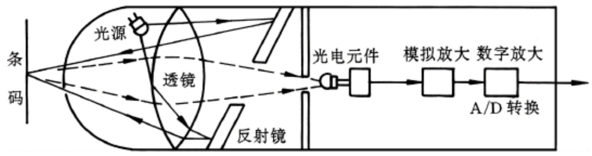


图 2　笔式阅读器结构原理

卡槽式阅读器与光笔原理相同。通常是将卡槽式阅读器安装于固定的位置,而将印有条码的卡片从卡槽中划过,读取条码信息。

也可以用图像传感器来阅读条码信息。这种条码阅读器不需要与条码之间有相对运动,只要将它轻轻靠近条码便可以迅速阅读出条码信息。它的可靠性高,价格也稍高。

用激光作为光源的条码阅读器称为**激光枪**。这种阅读器阅读速度高,可实现非接触式阅读,阅读器与条码之间的距离可达 500 mm,但价格昂贵。

参考文献

1. 胡嘉璋. 条码国家标准应用指南. 北京:中国标准出版社,1991

2. 黄志建,顾向阳,戴均陶. 条形码技术及应用. 北京:机械工业出版社,1992

(程天一)

tiaojie fangfa

调解方法(paramodulation method)　与**归结方法**结合使用能够高效地对含等词的**一阶逻辑**问题

进行推理。

相等关系在许多数学定理中均出现,因此需要研究含等词的一阶逻辑的高效推理方法。将数学定理用一阶逻辑表示时,相等关系用二元谓词"="表示。在一阶逻辑中"="只是普通谓词,为表达相等关系,一种方法是增加 $x=x$, $x=y\rightarrow y=x$, $x=y\wedge y=z\rightarrow x=z$, $x_j=x_0\wedge(P(x_1,\cdots,x_j,\cdots,x_n)\rightarrow P(x_1,\cdots,x_0,\cdots,x_n))$, $x_j=x_0\wedge(f(x_1,\cdots,x_j,\cdots,x_n)=f(x_1,\cdots,x_0,\cdots,x_n))$ 等公式来分别表达相等关系的反身性、对称性、传递性和等量替换性质(其中 P 和 f 是问题中出现的任何谓词和非常量函数, $1\leqslant j\leqslant n$)。这种方法的缺点是公式数目的增多使推理难度变大,因此另外一种方法是增加特定的推理规则来处理等词,比如调解方法。

1969 年 G. A. Robinson 和 L. Wos 提出调解方法。设 C_1 和 C_2 是无公共变量的子句,若 C_1 是 $L[t]\vee C'_1$, C_2 是 $r=s\vee C'_2$, $L[t]$ 是含项 t 的文字, C'_1 和 C'_2 是子句。若 t 和 r 存在最一般合一 σ,则称 $L^\sigma[s^\sigma]\vee C'^\sigma_1\vee C'^\sigma_2$ 为 C_1 和 C_2 的二元调解式, $L^\sigma[s^\sigma]$ 是将 L^σ 中的单独出现的一个 t^σ 代替以 s^σ 得到的文字。子句 C_1 和 C_2 的调解式 PR(C_1, C_2)是下面二元调解式中的一个:①C_1 和 C_2 的二元调解式;②C_1 和 C_2 的因子的二元调解式;③C_1 的因子和 C_2 的二元调解式;④C_1 的因子和 C_2 的因子的二元调解式。完备性定理:含等词的一阶逻辑子句集 S 是恒假的,当且仅当从 $S\cup\{x=x\}$ 出发,使用归结方法与调解方法可推出一对矛盾(即得到一个空子句)。

为提高调解方法的效率,在理论研究和实现方面均有大量工作,例如:引入项的序关系,调解只针对子句中该序关系下最大等词的最大的项进行。约束子句的方法(带有约束的调解):子句 C 附加约束 T,记为 $C|T$,表示 C 的全部基实例 C^σ,其中 σ 是 T 的解。若 T 不可满足则 $C|T$ 为重言式。在带有约束的调解方法中, $L[t]\vee C'_1|T$ 与 $r=s\vee C'_2|T'$ 的调解式为 $L[s]\vee C'_1\vee C'_2|T\wedge T'\wedge r=t\wedge \mathrm{OC}$,其中 $T\wedge T'$ 为继承下来的约束, $r=t$ 来存储合一约束,OC 表示推理步骤所要求的形如 $s>t\wedge\cdots$ 的有序关系约束。当推理步骤与约束不相容时该推理步骤被阻止。

针对特定等词理论的调解方法也被研究,1997 年 W. McCune 使用面向结合律与交换律的调解方法证明了 Robbins 猜想,这是第一个完全由自动定理证明器证明的数学猜想。

参考文献

1. 刘叙华. 基于归结方法的自动推理. 北京:科学出版社, 1994

2. Nieuwenhuis R, Rubio A. Paramodulation-based theorem proving. In: Robinson J A, Voronkov A, eds. Handbook of automated reasoning. Cambridge, MA: MIT Press, 2001, 371-443　(欧阳丹彤　冯莎莎)

tiaozhi jietiao qi

调制解调器(modem)　由调制器和解调器两部分组成,其中调制器将数据信息调制成适合传输信道传输的信号;而解调器则将接收的信号解调还原成数据信息。调制解调器属于**数据通信设备**中的**数据电路设备** DCE。

调制可分幅度调制(参见**幅度调制技术**)、相位调制(参见**相位调制技术**)、频率调制(**频率调制技术**)等三种基本方式。幅度调制也称为***幅移键控***(ASK),相位调制也称为***相移键控***(PSK),频率调制称为***频移键控***(FSK)。其中幅移键控最简单,但性能最差,易受干扰。频移键控对干扰不敏感,但要求信道的频带较宽。相移监控有两种基本方式:以载波相位为参考点的二进制相移键控(BPSK),以前一个码元的相位为参考点的差分相移键控(DPSK)。相移监控的信道频带比频移键控更宽,实现电路也更复杂,但其抗干扰能力最优。解调则与调制相反,从已调制的正弦波中分解出原来的数据信息。

常见的调制解调器包括:电话调制解调器、电缆调制解调器、无线调制解调器等。

电话调制解调器是计算机与电话线之间进行信号转换的装置,其中的调制器是把计算机的数字信号(如文件等)调制成可在电话线上传输的声音信号的装置;在接收端,解调器再把声音信号转换成计算机能处理的数字信号。通过调制解调器和电话线就可以实现计算机之间的数据通信。

ADSL(非对称用户数字线)modem 也是一种连接在电话线路上的调制解调器,和先前的电话调制解调器的主要区别在于,ADSL modem 不止局限在普通电话使用的语音载波的频段。常见的 ADSL modem 使用***编码正交频分***进行信号调制。

电缆调制解调器是另一种调制解调器,它是利用有线电视双向同轴网络提供互联网相关应用服务的技术。电缆调制解调器使用的是射频(RF)电视频道的一段载波频段。多个线缆 modems 可以使用一条电视电缆的相同频段,通过低水平介质访问协

议来实现在同一通道共同工作。典型的“上行”和“下行”信号用**频分复用**来隔离,它把用户要上传的上行数据以 5 ~42 MHz 的频率以 QPSK 或 16QAM 的调制方式调制之后向上传送,传输速率从 300 Kb/s 到 10 Mb/s。对于下行数据,解调的方式是 64QAM 或 256QAM,传输速率可达 40 Mbps。电缆调制解调器现存主要标准为由美国有线电视运营公司成立的行业组织 MCNS(多媒体线缆网络系统)起草的、被 ITU 批准的 J112 标准以及 IEEE 发布的 IEEE802.14 标准。

随着无线网络的普及,无线调制解调器也得到了越来越多的发展。用户可以通过无线调制解调器连接到无线网络中,而并非和电话调制解调器一样,直接连接到电话系统。移动电话或者 PDA 都可以用作为无线调制解调器而为计算机提供连接网络的无线接入点。在这种模式中,移动电话相当于一个沟通移动电话公司数据网络协议和计算机 Point-to-Point 协议(PPP)的网关。遵循 Wi-Fi 或者 WiMAX 标准的无线 USB 接口或串口调制解调器也可给计算机提供无线网络接入。还有一些无线调制解调器被制成 PCMCIA 或 CF 卡的接口,可以插入到计算机相应插槽上,提供无线网络连接的服务。

参考文献

1. Forouzan B A. 数据通信与网络. 2 版. 吴时霖, 周正康, 吴永辉,等译. 北京:机械工业出版社, 2002

2. Gilbert Held. 数据通信. 6 版. 戴志涛, 卞佳丽, 郑岩,译. 北京:人民邮电出版社, 2000

(王昊翔 张凌)

tiaozhi yu jietiao

调制与解调(modulation and demodulation) 调制(modulation)就是将信息源的信息作为控制信号对载波信号(carrier)进行处理,形成适合于特定信道传输的已调信号的过程。通常调制技术涉及载波信号的下列一个或几个参数的变化:幅度、频率和相位。解调(demodulation)是调制的逆过程。

一般来说,信息源的信息含有直流分量和频率较低的频率分量,称为基带信号(baseband signal,也称调制信号);利用基带信号进行直接传输的技术称为基带传输技术(baseband transmission technology);为了利用各种传输介质的有效带宽,通常可以将基带信号转变到一个较高的频率范围内进行传输,也称通带传输(passband transmission),这个转变过程称为调制,最终形成的已调信号也称通带信号。

调制可以根据信息源的类型,分为模拟调制和数字调制。例如:信息源为模拟广播和模拟电视,则可以通过控制高频载波信号的幅度、频率等,使其随着基带信号幅度的变化而变化来实现传统的收音机广播(AM,FM)和模拟电视机信号传输。现代计算机数据通信较多采用数字调制方法,例如数字有线电视采用数字调制技术在同轴电缆上传输数字电视和互联网数据。

调制也可以根据载波信号的类型,分为正弦波调制和脉冲调制等,调制的载波分别是正弦波、脉冲等。例如:正弦波数字调制有幅度调制(amplitude-shift keying, ASK)、频率调制(frequency-shift keying, FSK)和相位调制(phase-shift keying, PSK)三种基本方式以及目前常用的正交幅度调制(quadrature amplitude modulation, QAM)混合方式等。脉冲调制可分为脉冲幅度调制(pulse amplitude modulation, PAM)、脉宽调制(pulse width modulation,PWM)、脉位调制(pulse position modulation,PPM)和脉冲编码调制(pulse code modulation,PCM) 等方式。

参考文献

1. [日]关清三. 数字调制解调基础. 北京:科学出版社, 2002

2. 蒋青, 于秀兰. 通信原理. 北京:人民邮电出版社, 2008

(董守斌 张凌)

tingji wenti

停机问题(halting problem) 一个重要的**不可判定问题**,由它可以推出计算机科学中的许多不可判定问题。一般性的停机问题指是否存在这样的算法,使得对于任意的图灵机 M 和任意输入 x,可以判定对输入 x,图灵机 M 能否停机。早在 1936 年 A. Turing 证明了停机问题的不可判定性,他在证明一般性的停机问题的不可判定性时,采用了康托对角线化方法。对角线化方法已成为**递归论**和**计算复杂性理论**等学科中重要的工具,其中许多深刻的结论是通过这一方法获得的。由一般性的停机问题的不可判定性和通用图灵机的存在性,立即可推证特定图灵机的停机问题也是不可判定的。特定图灵机的停机问题指是否存在这样的算法,使得对于给定的图灵机 M 和任意的输入 x,可以判定对输入 x,图灵机 M 能否停机。停机问题中的另两个重要的不可判定问题是图灵机的完全性和等价性判定问题。图灵机的完全性问题指是否存在这样的算法,使得

对于任意图灵机 M,都可以判定 M 是否对任何输入都停机;图灵机的等价性问题指是否存在这样的算法,使得对于任何两个图灵机,都可以判定是否对任何输入这两个图灵机都停机,且有相同输出,或者都不停机。这类停机问题在程序设计学中均可找到实际的解释。

参考文献

1. 张鸣华. 可计算性理论. 北京: 清华大学出版社,1984

2. 李祥. 可计算性理论导引. 贵州人民出版社,1986

(马绍汉 李大兴)

tongxin shunxu jincheng

通信顺序进程(communicating sequential processes, CSP) 用于描述分布式通信系统的语言,首倡者是英国学者 C. A. R. Hoare。有两个不同而又相互联系的语言通常都称为 CSP。下面分别加以介绍。

1978 年 C. A. R. Hoare 提出的通信顺序进程 CSP,是面向分布式系统的程序设计语言。在该语言中,一个并发系统由若干并行运行的顺序进程组成,每个进程不能对其他进程的变量赋值。进程之间只能通过一对通信原语实现协作: $Q?x$ 表示从进程 Q 输入一个值到变量 x 中;$P!e$ 表示把表达式 e 的值发送给进程 P。当 P 进程执行 $Q?x$,同时 Q 进程执行 $P!e$ 时,发生通信,e 的值从 Q 进程传送给 P 进程的变量 x。后来出现的实用编程语言OCCAM即以 CSP 为基础发展而成。

1984 年 S. Brooks, C. A. R. Hoare 和W. Roscoe提出 CSP 理论(TCSP)。这是一个代数演算系统,其基本成分是事件(或动作)。进程由事件和一组算子构造而成。典型的算子有: →(前缀), |(外部非确定性选择), ⊓(内部非确定性选择), |||(交错并行), ‖(同步并行), \e(事件隐蔽),以及递归等。

例:(自动售货机)

VM = coin →(choc →VM | coffee →VM),

CUST = coin →(choc →CUST ⊓ coffee →CUST)

这里定义了两个进程: VM(售货机)和 CUST(顾客)。售货机在接受了硬币 coin 后,可按顾客的要求支付 choc 或 coffee。顾客在付了硬币后,或者想要 choc,或者想要 coffee,其选择不受外界影响。

与 CCS 不同(也与作为程序设计语言的 CSP 不同),TCSP 采用的是广播式通信,而不是握手式通信,即只有当并行运行的各进程都执行同一动作时,才发生同步。

TCSP 采用失败等价作为确定进程等价的准则,这也称为失败语义。一个失败是一二元组(s,X),其中 s 是事件的有限序列,X 是事件集(称为拒绝集)。若进程 P 可以执行事件序列 s,且到达一个无法执行 X 中任一事件的状态,则称(s,X)是 P 的一个失败。例如,设有进程 $P=a\to(b\to c\to \mathrm{STOP}|b\to d\to \mathrm{STOP})$,则$(ab,\{c\})$是 P 的一个失败,因为 P 可执行 ab,变成 $d\to\mathrm{STOP}$ 而无法做 c。类似地,$(ab,\{d\})$也是 P 的一个失败,但$(ab,\{c,d\})$不是。两个进程失败等价当且仅当它们具有相同的失败集。例如对自动售货机的例子,可以证明进程 VM | CUST 与 CUST 失败等价。

利用失败可以构造 TCSP 的指称模型,在此模型中,失败等价的进程被解释为同一个元素。关于失败等价建立了一些公理系统,可以对语义上的等价关系进行形式推导。

参考文献

1. 陆汝钤. 计算系统的形式语义. 北京:清华大学出版社,2017

2. Hoare C A R. Communicating sequential processes. Comm. ACM, 1978, 21(8): 666-677

3. Brooks S D, Hoare C A R, Roscoe A W. A theory of communicating sequential processes. J. ACM, 1984, 31(3): 560-599

(林惠民)

tongxin xitong yansuan

通信系统演算(calculus of communication systems, CCS) 英国学者 R. Milner 提出的用于描述通信并发系统的代数理论。

假定一个标号集 L,其补集 $\bar{L}\overset{\text{def}}{=}\{\bar{a}\mid a\in L\}$。$Act=L\cup\bar{L}\cup\{\tau\}$ 称为动作集,其中 τ 是特殊的不可见动作。CCS 的进程构造算子如下:

算子	直观意义
0	空进程
$a.P$	动作前缀($a\in Act$)
$P+Q$	非确定选择
$P\|Q$	并行复合
$P\backslash S$	限制($S\subseteq L$)
$P[f]$	换标号(f 是从 L 到 L 的部分函数)

此外还允许递归算子。

CCS 的语义由结构化操作语义方法给出,下面列出几条典型的语义规则。$P\xrightarrow{a}Q$ 表示进程 P 可执

行动作 a 而演变为 Q。

ACT $a.P \xrightarrow{a} P$ SUM1 $\dfrac{P \xrightarrow{a} P'}{P+Q \xrightarrow{a} P'}$ SUM2 $\dfrac{Q \xrightarrow{a} Q'}{P+Q \xrightarrow{a} Q'}$

PAR1 $\dfrac{P \xrightarrow{a} P'}{P|Q \xrightarrow{a} P'|Q}$ PAR2 $\dfrac{Q \xrightarrow{a} Q'}{P|Q \xrightarrow{a} P|Q'}$ COM $\dfrac{P \xrightarrow{a} P' \quad Q \xrightarrow{\bar{a}} Q'}{P|Q \xrightarrow{\tau} P'|Q'}$

RES $\dfrac{P \xrightarrow{a} P'}{P\backslash S \xrightarrow{a} P'\backslash S} a \notin S \cup \bar{S}$ REL $\dfrac{P \xrightarrow{a} P'}{P[f] \xrightarrow{f(a)} P'[f]}$

从关于并行算子|的规则可以看出，由两个进程并行复合而成的进程可以做每个分进程所能做的动作（PAR1）和（PAR2），但当两个分进程同时执行一对互补的动作时，则发生通信，产生 τ 动作（COM）。这种通信方式称为**握手式通信**。CCS 的基本思想是用 τ 和 + 来模拟|，将并发归结为非确定性，即所谓交错语义。这一思想体现在下面的展开律中：

$$Q_1|\cdots|Q_n = \sum\{\alpha.(Q_1|\cdots|Q_i'|\cdots|Q_n): i \leqslant n, Q_i \xrightarrow{\alpha} Q_i'\} + \sum\{\tau.(Q_1|\cdots|Q_i'|\cdots|Q_j'|\cdots|Q_n): i<j \leqslant n, Q_i \xrightarrow{l} Q_i', Q_j \xrightarrow{\bar{l}} Q_j'\}$$

CCS 采用**互模拟**作为基本的进程等价关系。强互模拟等价满足下面的 Monoid 公理：

$$P+0=P, P+P=P, P+Q=Q+P,$$
$$(P+Q)+R=P+(Q+R)$$

对观察等价（将 τ 忽略不计）还成立三条 τ-公理：

$a.\tau.P = a.P$,

$P+\tau.P = \tau.P$,

$a.(P+\tau.Q)+a.Q = a.(P+\tau.Q)$

上面介绍的 CCS 称为基本 CCS，或纯 CCS，进程之间只能通过执行互补动作实现同步，而不能直接进行通信。全 CCS 引入输出动作 $c!e$ 和输入动作 $c?x$，$c!e$表示沿通道 c 发送数据表达式 e 的值，$c?x$ 表示从通道 c 接收一个值赋给 x，此外还有条件表达式 if-then-else。用全 CCS 可以直接描述进程间的通信。

参考文献

1. 陆汝钤. 计算系统的形式语义. 北京：清华大学出版社，2017

2. Milner R. Communication and concurrency. New York：Prentice - Hall，1989

（林惠民）

tongyong fenzu wuxian xitong

通用分组无线系统（general packet radio service，GPRS）

在现有的全球移动通信系统（global system for mobile communications，GSM）基础上发展起来的一种移动分组数据业务，它介于第二代（2G）和第三代（3G）移动通信技术之间，因此常被称为“2.5G”。

在旧有系统中，一个通信过程的建立需要创建并保持一个电路连接，在整个通信过程中这条电路被独占直到该通信过程终止。区别于旧的电路交换方式，GPRS 将数据分成一定长度的包（分组），每个包的前面有一个分组头（其中的地址标志指明该分组发往何处），数据传送之前并不需要预先分配信道，而是在每一个数据包到达时，根据包头中的信息（如目的地址），临时寻找一个可用的信道资源将该数据包发送出去，也就是说多个用户可以共享一个相同的传输信道，每个用户只有在传输数据的时候才会占用信道。因此，GPRS 非常适合突发数据应用业务，如多播服务、短信和多媒体短信服务、因特网服务提供商服务、邮件服务等，这提升了 GSM 的数据服务性能。但由于当前 GSM 小区信道数量的限制、GPRS 系统本身编码方式导致无线信道传输速率较低等原因使得 GPRS 在大数据量业务应用中受限。

参考文献

1. 文志成. 通用分组无线业务——GPRS. 北京：电子工业出版社，2004

2. 钟章队，蒋文怡，李红君. GPRS 通用分组无线电业务. 北京：人民邮电出版社，2001

（张平）

tongyong jisuanji

通用计算机（general purpose computers）

为解决各种不同类型应用计算问题且具有良好通用性而设计的**计算机**。通用计算机功能齐全，具有较高的运算速度、较大的存储容量，配备较全的系统软件和工具软件、**输入输出设备接口**及通用外部设备。通用计算机适合于科学计算、数据处理、媒体处理、过程控制等多方面的应用，但其运行效率、速度和经济性依据不同的类型，不同的应用对象会受到不同程度的影响。

现代通用计算机普遍采用冯·诺依曼等人总结提出的存储程序体系结构，主要由输入设备、存储器、中央处理器和输出设备 4 个部分组成。按其规模、速度和功能等又可分为巨型机、大型机、**服务器**、**工作站**、微型机及单片机等。这些类型之间的基本区别通常在于其体积大小、结构复杂程度、功率消耗、性能指标、数据存储容量、指令系统和设备、软件配置等的不同。

随着**计算机网络**技术的发展和广泛应用，网络通信能力已成为评价现代通用计算机通用性的重要指标。

按通用计算的本质而论，通用计算机是指能够解决所有可表达为计算程序的问题的计算机，这些程序能够在有限的计算机存储容量、程序规模、运算速度及可靠性下执行。1934 年，阿兰-图灵（Alan Turing）证明了给定一个合适的程序，任何计算机的行为都可以用通用计算机来模拟。虽然其数学证明是纯理论的，但这个证明有着深远的意义。它说明了，若不考虑速度的因素，理论上任何现有的通用计算机都可以模拟未来可能出现的通用计算机。

拥有通用计算能力的计算机被称为是**图灵完全的**，这也常用来作为现代计算机的标准。1941 年德国研制的继电器式数字计算机 Z3，是最早具有图灵完全性的可运行的计算机（其证明在 1998 年完成）。其他早期的计算机，如美国 1942 年研制的电子管数字计算机 ABC、1943 年研制的机电式数字计算机 Mark I、1946 年研制的电子管数字计算机 ENIAC 等，都有可能是图灵完全的，都称为通用计算机。但这些计算机不具备实际可用的通用性，因此将它们视为通用计算机是不切实的。现代通用计算机不仅仅是理论上的通用，它们应该是非常实用的工具。虽然这些机器对现代通用电子数字计算机的发展具有很大的贡献，但其有限的通用问题解决能力使得它们的设计最终退出了历史舞台。

20 世纪 40 年代末，美国宾夕法尼亚大学摩尔电气工程学院设计了第一个存储程序计算机结构，该计算机结构被称为**冯·诺依曼体系结构**（von Neumann architecture）。由冯·诺依曼体系结构所定义的存储式程序，使得计算机能够充分利用其通用计算的性能。之后，多个研究机构采用冯·诺依曼体系结构最早研制出了存储程序式计算机，如 1949 年 5 月英国剑桥大学研制出的电子管存储程序计算机 EDSAC，1949 年 7 月美国研制的电子管存储程序计算机 BINAC 等。这些**存储程序式计算机**已配有汇编语言和系统管理程序，存储程序计算机硬件可以通过存储程序转变成各种不同的应用，已是具有实际可用性的通用计算机。他们的设计为现代通用计算机的发展奠定了基础。准确地讲，现代通用计算机都是基于二进制和存储程序的计算机。

20 世纪 50 年代末研制出的晶体管存储程序计算机，其运算速度比电子管存储程序计算机提高了近百倍，体积和功耗减少了几十倍。与此同时，程序设计语言及其编译器开始使用，系统管理程序的功能更加丰富，通用计算机产业已初具规模，使通用计算机不仅用于科学计算，还用于数据处理、事务处理和过程控制等。

20 世纪 70 年代以来，大规模集成电路和微处理器技术的迅速发展，使通用计算机的计算性能大大提高，体积和功耗显著减小，可靠性明显增强。特别是 20 世纪 80 年代后，精简指令集（RISC）微处理器的问世，使通用计算机的体系结构产生了重大变革，并行处理计算机体系结构技术的发展，使通用计算机的性能增长每年超过了 50%。与此同时，多种程序语言编译器、操作系统以及各种应用软件开发工具等软件技术随之迅速发展。通用计算机的通用本质得到充分体现和发挥，使现代存储程序计算机的存储程序转变非常高效，使计算机的使用越来越方便，使通用计算机的类型日益增多，应用领域越来越广泛。

参考文献

1. 赵明主编，史国川主审. 计算机应用基础. 北京：清华大学出版社，2009

2. Harver G c. Computer achitecter and impiemetation. Cabridge University Press，2000

3. 胡守仁. 计算机发展史（一）：早期的计算器与电子管计算机. 长沙：国防科技大学出版社，2004

（李思昆）

tongyong jicunqi

通用寄存器（general purpose register） 中**央处理器**内部用于在处理过程中临时存放中间结果和参数的一种多用途寄存器。它由与中央处理器其他逻辑电路同类的电路组成，运行速度快，可以被看作是比**高速缓冲存储器**速度更快的一个存储层次，但它不与存储器统一编址。

由于超大规模集成电路的发展，通用寄存器已由过去用若干个集成电路芯片所组成，发展为整体地集成在中央处理器芯片中。

通用寄存器用于存放操作数以及用作累加器、变址寄存器、基地址寄存器、自动增减量寄存器、栈指示器等。不同的计算机对通用寄存器的使用方式和配置数量可有很大差别。这些寄存器可以为各种用途共同使用，也可以有所侧重。但无论如何，涉及通用寄存器的指令必须包含寄存器的号码和使用方式码。前者用以选定某一个寄存器，后者指定该寄

存器的用途。有的指令不含使用方式码也必然有别的途径指明其用途。

寄存器与寄存器之间进行操作的指令运行速度最快,从这个意义上讲通用寄存器的数量越多越有利。但另一方面,过程调用时必须对现场进行处理,通用寄存器与存储器之间需要进行频繁的读写,从这个意义上讲通用寄存器的数量又不宜太多。另外,集成电路工艺水平也是制约通用寄存器配置数量的重要条件。目前多数计算机配置通用寄存器的数量在8~16个,这是考虑各种制约因素后的折中。

精简指令集计算机的结构比一般计算机简单,中央处理器芯片可以腾出足够的面积容纳较多的通用寄存器。有的精简指令集计算机利用这个条件配置了数以百计的通用寄存器,但每一个程序过程只能接触其中若干个寄存器,这些寄存器称为窗口。全部通用寄存器分为若干个窗口,一个过程使用一个窗口。相邻窗口有几个寄存器是相互覆盖的,这样就为沟通调用和被调用的两个过程之间的信息提供了方便。另外还有若干个寄存器存放全局性参数供各个过程共用。这样的使用方式大大减少了访问存储器的次数,使大多数操作都在寄存器与寄存器之间进行,从而提高了中央处理器的运行效率。

通用寄存器是从属于中央处理器体系结构的一个部件。随着电路集成度的不断提高和体系结构技术的改进,通用寄存器配置数量的制约将放宽,使用方式的设计自由度将扩大。 (张梓昌)

tongyong yidong tongxin xitong

通用移动通信系统(universal mobile telecommunications system, UMTS) 当前广泛采用的一种第三代(3G)移动通信技术,以宽带码分多址(wideband code division multiple access, WCDMA)接口技术为基础,由国际化标准组织第三代合作伙伴计划(3rd generation partnership project, 3GPP)制定。

UMTS结合了WCDMA的空中接口(移动电话和基站的空中通信协议)、GSM系统的移动应用核心部分(MAP)(此协议提供从用户或者到用户的呼叫路由功能)以及GSM的语音编码算法例如自适应多速率(AMR)和加强全速率(EFR)(它们定义了将语音数字化、压缩、编码的方法)。因此,UMTS能够兼容GSM,具有互操作性和全球漫游功能。尽管市场上现在的UMTS手机都是UMTS/GSM双模手机,但是它们都能在单一GSM网络中很好地工作。如果一个UMTS用户漫游到没有UMTS覆盖的地方,他的手机会自动切换到GSM模式;然而,普通GSM手机不能在UMTS网络使用。UMTS使用一对5 MHz带宽的信号,上行信道在1900 MHz附近,下行信道在2100 MHz附近,可支持高品质语音服务、3G视频电话服务、在线视频服务、网页浏览、电子邮件下载等多种应用业务。此外,UMTS使用相互认证及128位元加密机制,增强信息安全性。

世界上第一个UMTS网络2001年在马恩岛由Manx Telecom投入运营,随后欧洲、美国、日本的运营商们开始在全球内大范围部署UMTS网络。中国联通于2009年10月1日在国内285个城市正式提供UMTS服务。

参考文献

1. 苏信丰. UMTS空中接口与无线工程概论. 北京: 人民邮电出版社, 2006

2. 刘宝玲, 王莹, 陶小峰. UMTS网络技术. 北京: 电子工业出版社, 2005 (张平)

tongbu shuju lianlu xieyi

同步数据链路协议(synchronous data link protocol) 数据链路协议中的一种,它以定长的数据块为传输单位,称为帧(frame),在信道上传输。数据帧通常含有同步信息、检错信息等,因此同步协议能更有效地利用信道,也便于实现差错控制和高级的流量控制功能。同步协议又可分为面向字符的同步协议、面向比特的同步协议及面向字节计数的同步协议三种类型。

面向字符的同步协议

最早提出的同步协议,其典型代表是IBM的二进同步通信BSC(binary synchronous communication)协议,随后ANSI和ISO都提出了类似的相应标准。

任何链路层协议均可由链路建立、数据传输和链路拆除三部分组成。位实现建链、拆链等链路管理以及同步等各种功能,除了正常传输的数据块和报文外,还需要一些控制字符。BSC协议用ASCII和EBCDIC字符集定义的传输控制字符来实现相应的功能。数据报文一般由报头和文本组成。文本是要传送的有效数据信息,而报头是与文本传送及处理有关的辅助信息,报头有时也可不用。对于不超过长度限制的报文可只用一个数据块发送,对较长的报文则分作多块发送,每一个数据块作为一个传输单位。接收方对于每一个收到的数据块都要予以确认,发送方收到返回的确认后,才能发送下一个数

据块。

当发送的报文是二进制数据库而不是字符串时,二进制数据中形同传输控制字符的比特串将会引起传输混乱。为使二进制数据中允许出现与传输控制字符相同的数据(即数据的透明性),可在各帧中真正的传输控制字符(SYN 除外)前加上 DLE 转义字符,在发送时,若文本中也出现与 DLE 字符相同的二进制比特串,则可插入一个外加以标记。在接收端则进行同样的检测,若发现单个的 DLE 字符,则可知其后为传输控制字符;若发现连续两个 DLE 字符,则知其后的 DLE 为数据,在进一步处理前将其中一个删去。

由于 BSC 协议与特定的字符编码集关系过于密切,故兼容性较差。为满足数据透明性而采用的字符填充法,实现起来比较麻烦,且依赖于所采用的字符编码集。另外,由于 BSC 是一个半双工协议,它的链路伟传输效率很低。不过,由于 BSC 协议需要的缓冲存储空间较小,因而在面向终端的网络系统中仍然被广泛使用。

面向比特的同步协议

一种待传输的信息长度以比特为单位的数据链路协议,不受制于某个字符集。20 世纪 70 年代初,IBM 公司率先提出了面向比特的同步数据链路控制规程 SDLC;随后,ANSI 和 ISO 均采纳并发展了 SDLC,并分别提出了自己的标准:ANSI 的高级通信控制过程 ADCCP(Advanced Data Control Procedure),ISO 的高级数据链路控制规程 HDLC。链路控制协议着重于对分段成物理块或包的数据的逻辑传输,块或包由起始标志引导并由终止标志结束,也称为帧。帧是每个控制、每个响应以及用协议传输的所有信息的媒体的工具,所有面向比特的数据链路控制协议均采用统一的帧格式,不论是数据还是单独的控制信息均以帧为单位传送。每个帧前、后均有一标志码 01111110 用作帧的起始、终止指示及帧的同步。标志码不允许在帧的内部出现,以免引起歧义。为保证标志码的唯一性但又兼顾帧内数据的透明性,可以采用“0 比特插入法”来解决。该法在发送端监视除标志码以外的所有字段,当发现有连续 5 个“1”出现时,便在其后添插一个“0”,然后继续发后继的比特流。在接收端,同样监视除起始标志码以外的所有字段。当连续发现 5 个“1”出现后,若其后一个比特“0”则自动删除它,以恢复原来的比特流;若发现连续 6 个“1”,则可能是插入的“0”发生差错变成的“1”,也可能是收到了帧的终止标志码。后两种情况,可以进一步通过帧中的帧检验序列来加以区分。“0 比特插入法”原理简单,很适合于硬件实现。在面向比特的协议的帧格式中,有一个 8 比特的控制字段,可以用它以编码方式定义丰富的控制命令和应答,相当于起到了 BSC 协议中众多传输控制字符和转义序列的功能。作为面向比特的数据链路控制协议的典型,HDLC 具有如下特点:协议不依赖于任何一种字符编码集;数据报文可透明传输,用于实现透明传输的“0 比特插入法”易于硬件实现;全双工通信,不必等待确认便可连续发送数据,有较高的数据链路传输效率;所有帧均采用 CRC 校验,对信息帧进行编号,可防止漏收或重份,传输可靠性高;传输控制功能与处理功能分离,具有较大灵活性和较完善的控制功能。

面向字节计数的同步协议

该协议使用用户字符集中的一个特定字符划定帧的界限。这些协议大多数已被面向比特的协议取代。面向字节计数的同步协议的典型实例是 DEC 公司的数字数据通信报协议 DDCMP(Digital Data Communications Message Protocol)。由于采用字段计数方法来确定帧的终止边界不会引起数据及其他信息的混淆,因而不必采用任何措施便可实现数据的透明性,即任何数据均可不受限制地传输。

参考文献

1. Forouzan B A. 数据通信与网络. 2 版. 吴时霖,周正康,吴永辉,等译. 北京:机械工业出版社,2002

2. Gilbert Held. 数据通信. 6 版. 戴志涛,卞佳丽,郑岩,译. 北京:人民邮电出版社,2000

（许勇　张凌）

tongbu shuzi tixi

同步数字体系(synchronous digital hierarchy, SDH; synchronous optical network, SONET) 在传输媒质(如光纤、微波等)上实现同步信息传输、复用、分插和交叉连接的标准化数字信号结构等级和传送网技术体系。

SONET 是美国贝尔通信技术研究所(Bellcore)于 20 世纪 80 年代中期首先提出的用于光纤传输的物理层标准,后被美国国家标准协会(ANSI)标准化。1988 年国际电话电报咨询委员会(CCITT,国际电信联盟 ITU 的前身)采纳了 SONET 概念并适当改进后重新命名为 SDH,使其成为不仅适用于光纤也

适用于微波和卫星传输的通用技术体制。SDH 与 SONET 技术思想一致,只是在技术细节上略有差异。SONET 主要用于北美,SDH 主要用于欧洲、中国等其他地区。SDH /SONET 基于同步时分复用(TDM)技术,是对**准同步数字体系**(plesynchronous digital hierarchy,PDH)的一次革命,是传送网技术的一次重大进步。

SDH/SONET 帧结构是实现数字同步时分复用、保证网络可靠有效运行和全球标准化的基础。图 1 给出了 SDH 帧结构,SDH 采用块状帧结构,每帧由纵向 9 行和横向 270 × *N* 列字节组成,每个字节含 8bit。整个帧结构分成段开销(SOH)区、STM-*N* 净负荷区(payload)和管理单元指针(AU PTR)区三个区域,其中段开销区主要用于网络的运行、管理、维护及指配以保证信息能够正常灵活地传送,它又分为再生段开销(RSOH)和复用段开销(MSOH);净负荷区用于存放真正用于信息业务的比特和少量用于通道维护管理的通道开销(POH)字节;管理单元指针用来指示净负荷区内的信息首字节在帧内的准确位置以便接收时能正确分离净负荷。SDH 传输业务信号时各种业务信号要进入 SDH 的帧都要经过映射、定位和复用三个步骤,其中映射是将各种速率的信号先经过码速调整装入相应的标准容器(container),再加入通道开销(POH)形成虚容器(virtual containers,VC)的过程,帧相位发生偏差称为帧偏移;定位是将帧偏移信息收进支路单元(TU)或管理单元(AU)的过程,它通过支路单元指针(TU PTR)或管理单元指针(AU PTR)的功能来实现;复用是将多个低价通道层信号通过码速调整使之进入高价通道或将多个高价通道层信号通过码速调整使之进入复用层的过程。

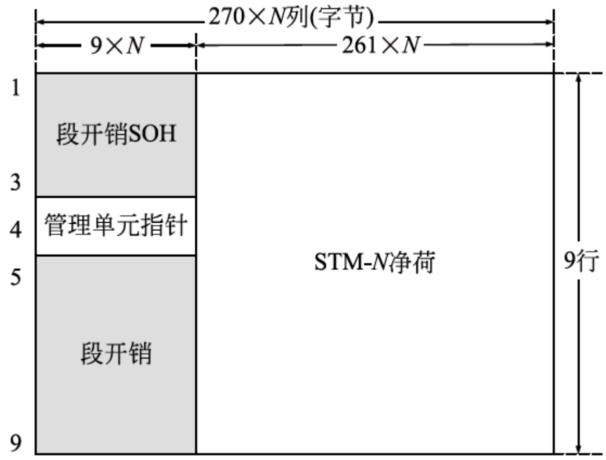


图 1　SDH 帧格式

SDH 采用的信息结构等级称为同步传送模块 STM-*N*(*N* = 1, 4, 16, 64),最基本的模块为 STM-1,由于帧周期为 125 μs, 即每秒传输 8000 帧,STM-1 的传输速率为 9 × 270 × 8 × 8000 = 155.520 Mb/s;SONET 也定义了类似的信息结构等级,称为同步传送信号 STS-*N*(*N* = 1,3,12,…,192),以及与光纤物理媒介相关的光纤载波 OC-*N*(*N* = 1,3,12,…,192)。SDH 中的 STM-1 与 SONET 中的 OC3(STS-3)的传输速率相同。表 1 给出了 SDH/SONET 中的信息速率等级。

表 1　SDH/SONET 速率等级

SONET 等级	帧格式	SDH 等级	帧格式	线路速率
OC-1	STS-1	—	—	51.840 Mbit/s
OC-3	STS-3	SDH-1	STM-1	155.520 Mb/s
OC-9	STS-9	—	—	466.560 Mb/s
OC-12	STS-12	SDH-4	STM-4	622.080 Mb/s
OC-18	STS-18	—	—	933.120 Mb/s
OC-24	STS-24	SDH-8	STM-8	1.244 160 Gb/s
OC-36	STS-36	SDH-12	STM-12	1.866 240 Gb/s
OC-48	STS-48	SDH-16	STM-16	2.488 320 Gb/s
OC-96	STS-96	SDH-32	STM-32	4.976 640 Gb/s
OC-192	STS-192	SDH-64	STM-64	9.953 280 Gb/s
OC-256	STS-256	—	—	13.271 040 Gb/s

续表

SONET 等级	帧格式	SDH 等级	帧格式	线路速率
OC-384	STS-384	—	STM-128	19.906 560 Gb/s
OC-768	STS-768	—	STM-256	39.813 120 Gb/s
OC-1536	STS-1536	—	STM-512	79.626 240 Gb/s
OC-3072	STS-3072	—	STM-1024	159.252 480 Gb/s

SDH 设备根据其用途可划分为终端复用器(TM)、再生中继器(REG)、分插复用器(ADM)和数字交叉连接设备(DXC),组网时可根据需要灵活合理地选择和配置各种设备,组成不同拓扑结构的网络。如利用 ADM 可组成 SDH 自愈环,自愈环的网络结构主要可分为以下四种,即单向通道倒换环、双向通道倒换环、二纤双向复用段公用保护环和四纤双向复用段公用保护环。SDH 自愈环有很强的保护/恢复性能,其保护/恢复时间能达到 50 ms 以内。由于 SDH 帧结构中安排了一定的开销比特用于维护管理,因此 SDH 网络具有十分强大的网络管理功能。

SDH 技术自从 20 世纪 90 年代引入以来,已经发展成为一种成熟、标准的技术,具有全球标准统一、组网灵活、上下电路方便、维护和管理功能强大、保护恢复能力强等优点,不但在骨干传输网中被广泛采用,在接入网中也得到了应用,已成为当今主流的光传送网技术。由于 SDH 是基于 TDM 的技术,很好地适应了话音业务的传输需求,但难以适应 IP 数据业务的突发性与速率可变性特点。随着业务的 IP 化,SDH 需要适应 IP 业务的要求进行改进和发展,以实现多业务承载。多业务传送平台(multi-service transport platform, MSTP)设备就是对传统 SDH 设备的继承和发展,是在 SDH 的基础上增强了数据业务处理功能,不但可以充分利用现有丰富的 SDH 网络资源、借鉴 SDH 系统多年的网络运维和管理经验、完全兼容目前大量应用的 TDM 业务,还可以实现 IP、以太网、ATM 等多种业务的综合传送和接入,满足日益增长的数据业务需求。

参考文献

韦乐平. 光同步数字传输网. 修订版. 北京: 人民邮电出版社, 1998

(唐雄燕)

tongyu

同余(congruence) 两个整数 a 与 b 之差若是自然数 m 的倍数,则称 a 与 b 模 m 同余。记为 $a \equiv b \pmod m$。这时 a 被 m 除所得的余数与 b 被 m 除所得的余数(在 0 与 $m-1$ 之间)相同。全体整数集合被分成 m 个互不相交的子集合,每个子集合中的数模 m 互为同余,不同子集合中的数模 m 不同余,称这样的子集合为剩余类。从每个剩余类中取出一个数作为该剩余类的代表,这 m 个数称为模 m 的一个完全剩余系。从两个剩余类中各取一个数,这两个数之和(差,积)所属的剩余类,称为这两个剩余类之和(差,积)。在剩余类之间有加法、减法和乘法,剩余类的集合成为一个环,称为剩余类环,记作 Z/mZ。零所属的剩余类为该环的零元素,1 所属的剩余类为该环中的乘法单位元。

一个剩余类中若有一个数与 m 互素,则该剩余类中所有的数都与 m 互素,称该剩余类与 m 互素。与 m 互素的剩余类的个数记为 $\varphi(m)$,称 $\varphi(m)$ 为欧拉函数。设 $m=p_1^{l_1}p_2^{l_2}\cdots p_s^{l_s}$ 为标准因子分解式,$p_1,\cdots,p_s$ 为互不相同的素数,则 $\varphi(m)=\varphi(p_1^{l_1})\varphi(p_2^{l_2})\cdots\varphi(p_s^{l_s})$,即 $\varphi(m)$ 为积性函数,而 $\varphi(p^l)=p^l-p^{l-1}$。从与 m 互素的每个剩余类中各取一数,这 $\varphi(m)$ 个数组成一个模 m 的缩剩余类。若整数 a 与 m 互素,则存在两个整数 x 与 y,使 $ax+my=1$,从而 $ax\equiv 1 \pmod m$,即 x 所属的剩余类是 a 所属的剩余类在乘法意义下的逆元素。所以与 m 互素的所有剩余类组成一个群。

当 $m=p$ 为素数时,模 p 的剩余类之间可以有加、减、乘、除(零元素不能作除数)的运算,Z/pZ 是一个包含 p 个元素的域。

整数 a 若与 m 互素,则 $a^{\varphi(m)}\equiv 1 \pmod m$ (费马小定理)。当 p 为素数时,有 $(p-1)!\equiv -1 \pmod p$ (Wilson 定理,其逆定理也成立)。

若 $f(x)=a_nx^n+\cdots+a_1x+a_0$ 为整系数多项式,求整数 a,使其适合 $f(a)\equiv 0 \pmod m$,称为求解同余方程。一次同余方程 $ax+b\equiv 0 \pmod m$ 有解的充分必要条件是 $(a, m) \mid b$。二次同余方程 $x^2\equiv a \pmod m$ 若有解,称 a 为模 m 的二次剩余,否则称 a 为模 m 的二次非剩余。

参考文献

华罗庚. 数论导引. 北京:科学出版社,1957

（裴定一）

tongji jiqi fanyi

统计机器翻译（statistical machine translation） 基于平行文本统计分析的一种**机器翻译**方法。统计机器翻译的基本思想是为翻译过程构建概率模型,通过对平行文本的统计分析估计模型参数,进而使用该模型进行翻译。

随着计算机运算能力的显著提升和**互联网**的高速发展,统计机器翻译在 20 世纪 90 年代以后取得迅速发展。相对于传统的基于规则的方法,统计机器翻译能够全自动从大规模真实文本中获取翻译知识,克服了传统方法面临的**知识获取**瓶颈问题。

统计机器翻译包含三个基本问题:

（1）建模　即为翻译过程建立概率模型。假设源语言句子是 f,目标语言句子是 e,$P(e|f)$ 表示将源语言句子 f 翻译成目标语言句子 e 的概率。早期的统计机器翻译基于噪声信道原理建立翻译模型。作为一种生成模型,噪声信道模型假定源语言中的句子 f 是由目标语言句子 e 经过含有噪声的信道编码后得到的。根据贝叶斯公式,噪声信道模型的解码取决于两个子模型:逆向的翻译模型 $P(f|e)$ 和语言模型 $P(e)$。逆向的翻译模型保证译文的忠实度,而语言模型则保证了译文的流利度。最早的翻译模型是 IBM 公司在 1993 年提出的基于词的模型。进入 21 世纪以后,以短语为基本翻译单元的基于短语的模型开始得到广泛研究。最近出现的基于句法的模型开始将句法信息引入翻译过程。另一类翻译模型是判别模型。与噪声信道模型不同的是,判别模型直接对条件概率 $P(e|f)$ 进行建模,噪声信道模型中的逆向翻译模型和语言模型都可以作为特征函数加入到判别模型中。目前,判别模型已经成为统计机器翻译的主流建模方法。

（2）训练　即从平行文本中自动估计翻译模型参数。统计机器翻译需要训练的参数分为两类:生成模型参数和判别模型参数。生成模型的参数一般是概率分布。估计生成模型参数分为两个步骤:首先抽取模型的参数实例,然后在训练语料库上估计概率分布。各种生成模型的抽取算法一般都利用词语对齐信息来降低复杂度。在概率估计方面,计算相对频度和期望最大化算法是最常用的两类方法。判别模型的参数一般是一组实数,即判别模型中特征函数的权重向量。统计机器翻译使用最广泛的判别模型参数训练算法是最小错误率训练算法。

（3）解码　即利用从平行文本中估计到的模型参数执行翻译。由于统计机器翻译为每个候选译文都赋予概率,解码的目标在于快速找到概率最大的译文。基于短语的模型一般采用语音识别中的栈搜索算法,将搜索过程中的局部翻译信息存放在栈序列中。基于句法的模型一般采用句法分析中的线图搜索算法,将搜索过程中的局部翻译信息存放在线图中。由于自然语言的复杂性,统计机器翻译解码算法通常面临指数级的搜索空间,需要采用剪枝策略来降低计算复杂度。直方图剪枝、阈值剪枝和立方体剪枝是统计机器翻译中常用的剪枝技术。

统计机器翻译的发展趋势是在保持全自动从平行文本中估计模型参数的同时逐步加深语言分析的层次。统计机器翻译的建模对象从词、短语发展到句法结构,未来有望将语义信息纳入统计建模中来进一步提高翻译质量。另外,当前的统计机器翻译研究主要集中在英语、汉语和阿拉伯语等少数几种语言。虽然统计机器翻译与具体语言无关,但是在设计翻译模型时需要考虑每种语言的具体特性。如何将统计机器翻译推广应用于全世界各民族语言是未来的重要研究方向。随着大规模分布式计算的普及,处理海量数据的并行机器翻译技术也将成为研究热点。

参考文献

1. Koehn P. Statistical machine translation. Cambridge: Cambridge University Press. 2010

2. Manning C, Schutze H. Foundations of statistical natural language processing. Cambridge, MA: MIT Press. 1999

3. 宗成庆. 统计自然语言处理. 2 版. 北京:清华大学出版社,2013

（刘洋　刘群）

tongji xuexi

统计学习（statistical learning） 以统计学为主要工具研究**机器学习**问题的一种有限样本归纳推理方法。一个完整的统计学习过程主要涉及统计学习问题的描述,统计学习算法的理论基础与性能分析方法,以及统计学习算法的设计与求解。作为一种有效的数据分析方法,统计学习的最终目标是更准确和更快速地解决更多的实际问题。

统计学习的基本理论体系形成于 20 世纪 70 年代,其中重要的进展包括 1968 年 V. N. Vapnik 和 A.

J. Chervonenkis 提出了刻画假设空间复杂度的概念——VC 维,1974 年得到了基于 VC 维与样本概率分布无关的依概率成立的学习机器泛化能力的界,并在此基础上提出了一种新的统计归纳原理——结构风险最小化原则。

相比于基础理论研究,统计学习算法的早期研究比较零散,且大多是启发式的,缺乏严谨的理论基础和统一的理论框架。统计学习算法取得重要进展的标志是 1992 年 B. E. Boser、I. M. Guyon 和 V. N. Vapnik 提出了具有统计学习理论依据的有效算法——SVM(Support Vector Machine)。

由于 SVM 在美国邮政服务数据库手写邮政编码的识别方面取得了成功,统计学习理论及其算法得到了计算机科学界的普遍关注。1995 年,V. N. Vapnik 出版了第一本关于统计学习理论的著作“The Nature of Statistical Learning Theory”,这是统计学习得到正式承认并进入飞速发展阶段的重要标志,V. N. Vapnik 是统计学习理论的创立者之一。对于统计学习,不同的研究者有着不同的观点和理解。统计学习方面另一本具有重要影响的著作是“The Elements of Statistical Learning: Data Mining Inference and Predication”。

自 1995 年以来,统计学习不断向广度和深度发展,一些新的统计学习范式不断涌现,一些新的统计学理论分析方法、算法设计原则和优化方法不断被引入。统计学习已经进入计算机科学的不同领域甚至其他学科,成为很多学科或者交叉学科不可或缺的支撑技术。总之,数据的复杂性、需求的多样性以及数据量的日益增长是推动统计学习发展的主要动力。

一个统计学习问题由四个部分组成,即样本集合、模型空间、代价函数和目标函数。样本集合是样本空间的有限子集,由独立同分布的一些样本组成,但它们的概率分布函数是未知的,这些有限的样本蕴涵了一定的统计规律,构成了机器学习问题的示例。样本集合有时也称为训练集合。模型空间由定义在样本空间上的一些函数组成,它限定了统计学习解的求取范围,统计学习的最终结果为模型空间中满足一定条件的某个函数。对不同的统计学习问题如何设定相应的模型空间称为模型选择问题。模型有时也称为一个假设,假设空间指的就是模型空间。代价函数是关于样本与模型的一种非负函数,是对模型决策一个样本时所付出代价的一种度量,对某一样本和模型,如果代价函数为零,说明模型对该样本采取了完全正确的决策。代价函数有时也称为损失函数。目标函数是定义在模型空间上的函数,是一个模型对样本空间所有样本决策时的一种总体平均代价,是对模型总体决策性能的一种度量,也称为该模型的期望风险。它可以用关于样本概率分布函数的黎曼-斯蒂尔切斯(Reimann-Stieltjes)积分表示。

对一个统计学习问题,一个模型所对应目标函数值的大小定量描述了该模型对样本空间所有样本的适用性,直接体现为其泛化能力的强弱。泛化能力强的模型,预测能力自然就越强。统计学习的目标是在模型空间找到泛化能力最强的模型,即使目标函数达到最小的模型,这个模型称为该统计学习问题的解。由此可知,统计学习是在不知道概率分布函数的情形下,仅仅根据样本空间的子集(即训练集合)就试图归纳推理得到在样本空间上泛化能力最强的模型。但也正是由于样本的概率分布函数未知,统计学习问题目标函数的解析形式无法得到,因此统计学习问题的解在实际中是无法求得的。

给定一个统计学习问题,任何试图求得该问题近似解的算法都可以称为一种机器学习算法,但是否具有统计学习理论依据或者真正称之为统计学习算法还需要进一步的分析和讨论。为了有效求解,统计学习算法在形式上往往表现为在样本集合上遵循某种准则。这种准则显然不同于统计学习的目标函数,它完全用由样本集合而不是整个样本空间确定。统计学习算法的目标是在模型空间找到使这种准则达到最优的模型,这个模型称为该统计学习算法的解。不同于统计学习问题的解,由于样本集合已知,统计学习算法的求解一般是可以进行的。特别,当统计学习算法所遵循的准则表示为关于样本集合的优化问题时,这个优化问题的目标函数就是准则函数,优化问题的解就是统计学习算法的解。

由于统计学习算法的解仅仅建立在样本集合的基础上,它与统计学习问题的解不可避免地存在着一定的差异。即使是对于同一个统计学习问题,也存在有多种不同的统计学习算法。为了衡量不同统计学习算法解的优劣,一般将原有的样本集合进一步拆分为新的样本集合和测试集合,根据模型在测试集合上表现来评价统计学习算法解的性能。

统计学习研究的基本内容主要包括:

(1)研究学习过程的一致性,即统计学习算法的解与统计学习问题的解是否一致的问题。学习过程的一致性是衡量学习算法是否具有统计学习理论

依据的关键标准之一。在 V. N. Vapnik 统计学习理论中,使用依概率近似正确(Probably Approximately Correct)来描述有限样本学习过程的一致性,这种描述也称为 PAC 学习模型,PAC 学习模型由 L. G. Valiant 提出。

对一个特定的统计学习算法,学习过程的一致性主要体现在以有限训练数据为准则统计学习算法的解能否在样本数目趋于无穷大时依概率逼近统计学习问题的解。在 V. N. Vapnik 统计学习理论中,依概率成立的与样本概率分布无关学习机器泛化能力的界占据极其重要的地位,它能够合理描述统计学习算法解与统计学习问题解之间的差异,这种界是 PAC 模型在具体学习过程中的一种体现。因此,学习过程的一致性问题进一步转化为分析当样本数目趋于无穷大时学习机器泛化能力的界是否趋于 0。

(2) 研究学习过程的收敛速度,即定量描述统计学习算法解与统计学习问题解之间差异与样本数目的关系。学习过程的收敛速度是衡量学习机器性能的重要标准之一。学习过程的收敛速度越快,从理论上说,其达到设定精度统计学习问题的解所需要的样本数目就越少,这种学习算法有限样本归纳能力就越强,性能也就越好。在 V. N. Vapnik 统计学习理论中,主要使用与概率分布无关学习机器泛化能力的界来描述学习过程的收敛速度,学习过程的收敛速度体现为泛化能力界的大小。

(3) 研究学习算法的设计问题。如何在学习过程的一致性和收敛速度的基础上得到样本集合上的准则,克服过拟合(overfitting)问题,进而对这种准则进行优化以得到统计学习算法是学习算法设计首先必须解决的问题。其次,统计学习算法除了要求遵循学习过程的一致性和具有理想的收敛速度外,由于实际问题的需要,学习算法的解还需要具有一定的结构信息。如何面向实际需求,在样本集合上设计多样化含有结构信息的准则从而获得兼顾结构和总体平均代价的解也是统计学习算法设计需要解决的问题。另外,算法设计还要从理论和实验等方面研究算法的有效性,其中包括研究算法对训练样本隐含的假设和实际应用的效果,从而为具体学习问题使用何种算法提供指导。

(4) 研究学习算法的求解问题。统计学习正面临着数据规模日益增长的严峻挑战,如何处理大规模甚至超大规模的数据是统计学习算法求解首先需要解决的问题。其次,在保证泛化能力的基础上,如何更快地获得学习算法的近似解是需要解决的另一个重要问题。另外,如何求解统计学习导致的新优化问题以及在求解过程中如何有效保证统计学习问题的结构也是算法求解需要解决的问题。

统计学是统计学习重要的理论基础之一。统计学为统计学习提供了多种形式的理论分析方法。实际上,统计学习问题是一个非常一般的科学问题,统计学中所研究的几乎所有问题都可以在统计学习中找到其相对应的部分,一些重要的结论也是首先在学习理论的范畴内被发现,然后使用统计学的术语重新表达。但统计学所研究的主要是渐近理论,即当样本数目趋于无穷大时的统计性质。而在统计学习理论中,更加强调了有限样本统计学的问题。统计学习理论的创始人 V. N. Vapnik 建议统计学习的研究应该遵循一个原则,即在解决一个给定学习问题时,要设法避免把解决一个更一般的问题作为其中间步骤。另一方面,统计学习又是面向实际问题的,在有限样本归纳原理的基础上,统计学习算法的合理性往往由具体的学习效果确定。

优化理论及其算法在统计学习中占据着越来越重要的地位。优化算法是求解统计学习问题的主要手段,各种数值优化算法的引入极大地丰富了机器学习问题求解的研究。但来源于优化理论的方法重点关注的是算法的收敛速度与精度,而在统计学习问题中,由于只有有限样本,且实际训练数据来源存在不确定性和噪声,同时统计学习算法的解与统计学习问题的解之间不可避免地存在着一些差异,因此统计学习的优化算法无须具有很快的理论收敛速度或者以很高的精度收敛,只要能很快获得较好泛化性能的模型且计算开销小即可。另外,统计学习的发展又对优化理论及其算法提出了新的挑战,如何有效求解一些新的和大规模统计学习优化问题以及如何保证解的结构已经成为当今优化领域强烈关注的问题。

参考文献

1. Vapnik V N. The nature of statistical learning theory. New York: Springer-Verlag, 1995

2. Hastie T, Tibshirani R, Friedman J, Franklin J. The elements of statistical learning: Data mining, inference and prediction. New York: Springer-Verlag, 2009

3. Boser B E, Guyon I M, Vapnik V N. A training algorithm for optimal margin classifiers. In: Haussler D, ed. Proc 5th Annual ACM Workshop Comput-

er. Learning Theory. 1992. 144-152

4. Valiant L G. A theory of the learnable. Communications of the ACM, 1984, 27(11): 1134-1142

（陶卿）

tongyi jianmo yuyan

统一建模语言（unified modeling language, UML） 一种基于反映多种面向对象建模方法统一的可扩充建模语言。1997年统一建模语言（UML）被美国对象管理组织（OMG）采纳为该组织的建模语言规范。UML主要用于软件密集型系统的规约书写、系统构作、文档形成，并力求形象直观。它定义了建立系统模型所需的概念并给出其直观表示法，但并不涉及如何进行系统建模。因此，它只是一种建模语言，而不是一种建模方法。

UML的演化历史可概括为四个阶段。最初阶段是面向对象方法学家的联合行动，由G. Booch, J. Rumbaugh和I. Jacobson将他们各自的方法结合起来，形成统一方法0.8和0.9。第二阶段是公司的联合行动，有十多家公司组成UML伙伴组织，共同提出UML1.0和UML1.1，作为向OMG提交的方案。第三阶段是在OMG的组织下进行修订和改进，产生了UML1.2、UML1.3和UML1.4等版本。目前所处的阶段是进行较为重大的修订。不久前已推出UML 2.0版本。

基本成分

统一建模语言（UML）的基本成分是图与建模元素。UML定义了9种用于建立系统模型的图以及为各种图所公用的建模元素和扩展机制。图有类图、对象图、用案（或称用况）图、顺序图、协作图、状态图、活动图、构件图、部署图。其中类图和对象图统称静态结构图；顺序图和协作图为两种不同的交互图；构件图和部署图均为实现图。建模元素有串、关键词、表达式、包、子系统等，并给出了其表示符号。扩展机制有约束、注释、标记值以及衍型。这些元素可以添加到其他建模元素之上，从而将原来的建模元素特化为一种语义较为特殊的新变形，或者表示出它们的某些细节。这样，将起到对UML扩充或细化的作用。

（1）静态结构图——类图和对象图　UML的类图是诸如类、接口及其关系等静态说明的模型元素集合，用于展示模型之静态结构。和大部分面向对象建模方法不同的是，UML类图中作为结点的元素不只是类，也可以是接口、类型、包，甚至可以是实例。对象图是由实例构成的图形，包括对象和数据的值；UML称静态对象图为类图的实例，它显示了在一个时间点上系统细节状态的一个快照。

（2）用案图　用于表现参与者、用案及其间的关系。其中用案是由系统与一个或多个外部交互者（即参与者）之间交换的消息序列以及系统执行的活动共同体现的。参与者定义为一个实体的使用者在与该实体交互时所扮演的角色的集合。

（3）顺序图　是UML提供的两种交互图之一，它展示按时间顺序排列的对象交互。特别是，它通过对象"生命线"来展示参加交互的对象，并按时间顺序展示它们之间所传送的激发。它所展示的交互是在一个协作中，对象之间为产生所要求的操作或结果而交换的一组信息。

（4）协作图　是UML的另一种交互图，它表示一个协作，其中包含在一个特定上下文中由对象扮演的一组角色以及它们所需之关系。协作图也可以表示交互，它定义了一组消息，这些消息说明了在协作中扮演角色的对象之间为达到所需的结果而进行的交互。和顺序图不同的是，协作图表现的是对象角色之间的关系，并不表示时间顺序。

（5）状态图　通过指明一个实体对于接收到的事件之反应而表现该实体的行为，特别是动态行为。

（6）活动图　是状态图的变形，其中状态表示动作或子活动的执行，而转换是由动作或子活动的完成而触发的。它表现为一个过程本身的状态机。活动图用于对系统的行为建模。

（7）实现图——构件图和部署图　是两种表现系统实现问题的图。其中构件图表现系统将实现的是哪些软件构件以及这些构件之间的依赖关系。构件是系统中一个符合一定标准、可部署、可替换、封装了实现而显露一组接口的部件，可部署到处理机结点上。部署图显示了运行时的处理单元以及在其上执行的软件构件、进程和对象所共同构成的配置。

新近发展

UML的出现使面向对象建模概念和表示法趋于统一和标准化。它应用广泛、成效显著，引起工业界和学术界的普遍重视。然而，UML也存在不足之处。例如，内容庞大，复杂无理，延拓或缺，严谨欠佳等。为此，OMG自1999年起即酝酿UML的下一次重大发布，计划进行较大的实质性变动，以推出UML 2.0。经过数年的准备，UML 2.0已在不久前通过。其中虽有不少变动，但是否属实质性的变动，尚有待业者各抒己见。

参考文献

1. Fowler M, Scott K. UML distilled: a brief guide to the standard object modeling language. 2nd ed. Addision-Wesley Longman Inc., 2000(中译本:UML 精粹. 2 版. 标准对象建模语言简明指南. 徐家福,译. 清华大学出版社,2002)

2. Kobryn Cris. UML 2001: a standardization odyssey communication of the ACM. Oct. 1999, 42(10)

3. OMG Unified Modeling Language Specification Version 1.4. Spet. 2001. http://www.omg.org/cgi-bin/doc? formal/ (邵维忠)

tongyi ziyuan dingwei dizhi

统一资源定位地址(Uniform Resource Locator, URL) 互联网中信息资源的访问地址,简称 URL 地址。信息资源可以是存储在服务器中的各类文件,包括:网页、图形、声音、视频、文件以及应用程序等。URL 地址由三个部分组成:①所使用的传送协议;②文件所在服务器的域名或 IP 地址;③文件路径名,可选。如果第一部分为 http://,则为**超文本传送协议**,用于访问网页;如果是 ftp://,则为**文件传输协议**,用于访问 ftp 文件。URL 的第一部分也可以是 https, gopher, telnet, wais, news, ldap, mailto 等协议。例如,URL 为 http://www.abc.com/home/news.html,指出浏览器使用 HTTP 协议在域名为 www.abc.com 的 Web 服务器上访问路径名为/home/news.html 的网页文件。如果没有给出文件路径名,URL 引用路径中最后一个目录中的默认文件名,一般为 index.html 或 default.htm,这个文件通常为网站的主页。

URL 可以直接在**浏览器**中输入,也可以通过单击网页中的链接进行访问。

参考文献

Encyclopedia Britannica online, 2014 (张蓓)

toukui xianshiqi

头盔显示器(head mounted display, HMD) 一种固定在头部或头盔上,为使用者提供可视的视频图像以及字符信息的小型显示装置。主要应用在航空飞行领域,以及诸如游戏、体育、医疗、虚拟训练与仿真等其他领域。

典型的头盔显示器具有一至两个微型显示单元,这些显示单元可能是彩色阴极射线管(CRT)、液晶显示器(LCD)、反射型液晶元件(liquid crystal on silicon, LCOS),甚至是有机发光二极管(Organic Light Emitted Diode, OLED)。显示器上可以显示计算机生成的图像,也可以把实际场景和计算机生成的图像叠加进行显示。高级的头盔显示器可以根据头部、手或眼球的运动而改变显示的图像。

头盔显示器一般具有的功能有:①将图像和文字信息全部呈现在佩戴者前面的视场内,使其能集中精力进行操作。由于佩戴者的眼睛不需要反复聚焦,因此消除了眼睛的疲劳。②利用附带的头部跟踪器,将实时获取的头部位置和运动信息传递给生成图像的计算机,计算机就可以根据佩戴者的位置和方向信息提供不同图像画面。

头盔显示器一般由以下部分组成:头盔、头盔跟踪器、电子组件、控制组件、计算机处理系统、发射/接收机、头盔矫正器、显示部件等。其中头盔主要起支撑作用;头盔跟踪器主要作用是跟踪佩戴者头盔,以确定其头部的确切位置和视角;计算机处理系统用来处理来自各信息源的大量数据、图像信息;显示部件显示由符号、数字、图像合成的信息。

按照使用类型,头盔显示器大体分为:航空用机载头盔显示器、地面车辆用的头盔显示器以及其他医学、游戏、体育、模拟训练、工程领域使用的头盔显示器。

设计头盔显示器需要考虑的因素有:能否显示立体图像、瞳间距、视野的大小、分辨率大小、远程聚焦能力、处理单元和操作系统等。

作为**虚拟现实**技术关键设备的头盔显示器,其未来的发展方向归纳起来有如下方面:

(1) 双目立体显示技术 新一代头盔显示器将是高性能双目立体头盔显示器。

(2) 头部跟踪技术 头部跟踪技术研究包括探索新的瞄准线探测方法、提高探测精度、扩大头盔瞄准线的探测范围、研究先进的数字处理方法及预处理技术、提高头部跟踪系统的采样速度以及处理能力等。

(3) 头盔和微显示器技术 采用更轻、更耐用的头盔材料和更小更轻的元器件来改进现有头盔;采用全息光学元件和二元光学元件代替传统光学元件,以利于优化系统,减轻重量。另一方面小尺寸、高分辨率、高亮度、低功耗的小型平板显示器亦是今后研究的方向。

参考文献

1. 周海宪. 头盔显示技术的发展. 红外技术,

2002, 24(6)

2. 吴卫玲. 头盔显示器的技术发展综述. 科技信息, 2010(33)

（李国宽）

touyingyi

投影仪(projector)　将计算机或其他信息设备输出的文字、图像和视频信息通过光学投影原理进行放大,投射到屏幕或墙面上以显示信息的装置。投影仪又称投影机。

投影仪的主要技术指标有输出分辨率、光输出、对比度和均匀度等。输出分辨率用横向像素和纵向像素的乘积来表示,如 800×600,1024×768,1920×1080 等。光输出指投影机输出的光能量,单位为流明(lm)。一般便携式投影仪的光输出在 1000～3000 lm,高亮投影仪的光输出可达 10 000 lm 以上。与光输出有关的一个物理量为亮度,是指屏幕表面受到光照射的光能量与屏幕面积之比,亮度常用的单位是勒克斯(lx, $1\ lx = 1\ lm/m^2$)。对比度是屏幕上同一点最亮时(白色)与最暗时(黑色)的亮度的比值,目前大多数 LCD 投影仪的标称对比度都在 400∶1 以上,而大多数 DLP 投影机的标称对比度都在 1500∶1 以上。如果用来演示色彩丰富的照片和播放视频动画最好选择 1000∶1 以上的对比度。均匀度是指投射光的亮度在整个屏幕上的均匀程度。

投影仪主要有 CRT(阴极射线管)、LCD(液晶显示器)、LCOS(硅基液晶)、DLP(数字光路处理器)和 DLV(数字光阀)等五种类型。CRT、LCD 和 DLV 投影仪采用透射式投影方式,LCOS 和 DLP 采用反射式投影方式。

(1) CRT 投影仪　CRT 投影仪也叫三枪投影仪,其工作原理与 CRT 显示器一样,由阴极射线电子束扫描击射在成像面上,使成像面上的荧光粉发光形成图像后,再传输到投影面上。和 LCD 投影仪相比,它的分辨率高、对比度好、色彩饱和度佳、对信号的兼容性强,且技术十分成熟。但由于 CRT 投影仪体积大、亮度低,目前已很少使用。

(2) LCD 投影仪　LCD 投影仪的主要成像器件是液晶板。与液晶显示器相同,LCD 投影仪采用的是扭曲向列型液晶。LCD 投影仪的像素是液晶板上的液晶单元。LCD 投影仪内部有红、绿、蓝三片液晶板分别作为红、绿、蓝三色光的控制层。光源发射出来的白色光经过镜头组后会聚到分色镜组,红色光被分离出来,投射到红色液晶板上,液晶板“记录”下的以透明度表示的图像信息被投射生成了图像中的红色光。绿色光被投射到绿色液晶板上形成图像中的绿色光,同样蓝色光经蓝色液晶板后生成图像中的蓝色光,三种颜色的光在棱镜中会聚,由投影镜头投射到投影幕上形成一幅全彩色图像。LCD 投影仪体积较小、重量较轻,制造工艺较简单,亮度可达 6000 lm(2011 年),分辨率可达 2500×2000(2011 年),是目前市场上占有率最高、应用最广的投影仪。

(3) LCOS 投影仪　LCOS 与 LCD 投影的基本原理类似,不同之处在于利用光的反射而非透射。当光线照射到 LCOS 芯片上时,反射光受 CMOS 电极和含有透明电极的玻璃(ITO)极板间的电场调制,将图像或数据信息转换为互补金属-氧化物-半导体(CMOS)电极阵列上的电压排布,进而控制液晶的开关实现反射光成像。

(4) DLP 投影仪　DLP 投影仪是一种光学数字化反射式投射设备。DLP 投影仪的关键是数字微反射成像器件 DMD。一片 DMD 是由许多个微小的正方形反射镜片(简称微镜)按行列紧密排列在一块硅晶片上,每一个微镜都对应着一个存储器,存储器可以控制微镜在 ±10°角两个位置上切换转动。其基本原理是,光束通过一高速旋转的三色透镜后,再投射在 DMD 部件上,生成图像的一个像素。当微镜在某一个位置时,光线通过光学透镜投射在大屏幕上完成图像。DLP 投影仪可以产生清晰度高、画面均匀、色彩还原性好的图像,分辨率达到 1280×1024(2011 年)。其光学路径相当简单,体积小,重量轻。主要应用在便携式系统中。

(5) DLV 投影仪　液晶光阀投影仪是 CRT 投影仪与液晶光阀相结合的产物。CRT 显像管成像后的图像送到光电转换器,形成的电信号用来控制液晶。高强度的光源将液晶形成的图像投射在大屏幕上。其亮度可达 6000 lm,分辨率则可达 2500×2000(2011 年)。这类投影仪非常适合在光线较强、观众较多的场合中使用。

投影仪的光源有高压汞灯、发光二极管(LED)、激光等几种。高压汞灯亮度高,但寿命不长(一般为几千小时),失效后需换灯泡,造成使用成本增加。LED 作为光源具有节能和长寿命(几万小时)的特点,但亮度不够高,一般用于小型和微型投影仪中。激光光源亮度高、寿命长,但价格昂贵,且有散斑现象。也有将激光和 LED 结合起来的混合光源,价格适中,并具有高亮度和长寿命的特点。

投影仪在教学、讲演、商业展示、视听娱乐等方

面的应用十分广泛，特别是便携式投影仪和微型投影仪的发展十分迅速，手持式设备如部分数码相机、家用摄像机和手机已开始使用附带的微型投影部件来投射显示影像信息。

参考文献

张景生. 投影机使用与维修. 北京：国防工业出版社，2010 （林兼 谢长生）

tuling guiyue

图灵归约（Turing reduction） 一种应用范围广泛的**复杂性归约**。其严格的形式定义要使用 oracle 图灵机。所谓 oracle 图灵机是一个多带图灵机 M，它有一条特殊的工作带，叫做 oracle 带或询问带，和三个特殊的状态：询问状态 $q_?$ 以及回答状态 q_Y 和 q_N。Oracle 图灵机 M 的计算依赖于事先给定的 oracle 集 B。当 M 处于询问状态 $q_?$ 时，若 oracle 带上的字符串 $u \in B$，则 M 转移到状态 q_Y；若 $u \notin B$，则 M 转移到状态 q_N。无论是哪一种情况，在这一步 oracle 带都被清成空白带，而其他的带保持不变。这样一步计算称为询问。除询问之外，oracle 图灵机的计算和普通图灵机的计算相同。用 M^B 表示以 B 为 oracle 的 oracle 图灵机 M^B 接受的集合（语言）记作 $L(M^B)$。

设 A 和 B 是两个集合（语言）。如果存在 oracle 图灵机 M 使得 $A = L(M^B)$，则称 A 可图灵归约到 B。如果这个 M 是确定型多项式时间的 oracle 图灵机，则称 A 可多项式时间图灵归约到 B。 （张立昂）

tulingji

图灵机（Turing machine） 英国数学家 A. M. Turing 于 1936 年提出的一种理想的计算机器的数学模型，现在已成为计算机科学中**可计算性理论**和**计算复杂性理论**的基础。

图灵机分为确定型与非确定型两大类，每类中又主要有单带、多带等许多形式与变种（已证明其计算能力是等价的）。一台标准的确定型单带图灵机由一条双向可无限长的被分为一个个小方格的磁带、一个有限状态控制器与一个读写磁头构成。图灵机一步步地进行工作，机器工作情况取决于三点：首先是机器的内部状态，其次是读写磁头扫描在磁带的哪个方格上，再次是读写磁头扫描着的方格上有什么信息。机器执行一步工作是如下进行的：读写磁头在所扫描的方格上写上符号（原有的符号自然消除），磁头向右或向左移动一个方格，机器由当前所处的状态转向另一个状态，然后进行下一步工作；如此周而复始地机器一步一步工作，除非遇到某一命令机器停止工作的状态，否则不停止。例如，图 1 中的机器在某一步上处于状态 q_3，将所扫描方格中的 A 改写为 E，左移一个方格，进入状态 q_5，此时机器磁头扫描方格内的字母为 T；机器下一步的工作现在完全由状态 q_5 和扫描方格内的信息 T 唯一确定。

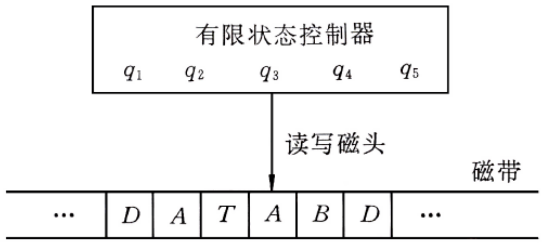


图 1 图灵机

图灵机的程序确定了图灵机的这种由状态、符号决定的一步一步的工作。有许多种方法来定义一台图灵机的程序（例如：流程图，伪汇编语言等），通常方便的方法是由下述五元组所确定的一个阵表来定义一台图灵机：

$$\langle q, A, E, M, q' \rangle$$

其中 q, q' 表示有限状态控制器中的状态；A, E 表示磁带方格上的符号，M 表示 L（左移）、R（右移）或 N（不动）；这样，上面例子中图灵机的一步动作可用五元组

$$\langle q_3, A, E, L, q_5 \rangle$$

来确切地描述。下述阵表是一个计算函数 $f(x) = 2^x$ 的图灵机程序，其中 B 代表磁带上的方格为空白方格，此外，一些约定如下：

（1）x 和 $f(x)$ 的值以二进制表示；

（2）在开始时，磁带上只有一连续的方格串上放入相应于 x 的二进制值，其余方格均为空格；

（3）机器从状态 q_1 开始，磁头扫描在 x 最左位所在的方格上；

（4）停机时，$f(x)$ 的值就是磁带上非空方格所组成的二进制串。

计算 $f(x) = 2^x$ 的图灵机程序如表 1 所示。

表 1 计算 $f(x) = 2^x$ 的图灵机程序

当前状态	被扫描时的写、移动、状态转移		
	B	0	1
q_1	$1, L, q_7$	$0, R, q_1$	$1, R, q_2$
q_2	B, R, q_3	$0, R, q_2$	$1, R, q_2$

续表

当前状态	被扫描时的写、移动、状态转移		
	B	0	1
q_3	$0,L,q_4$	$0,R,q_3$	Error
q_4	B,L,q_5	$0,L,q_4$	Error
q_5	Error	$1,L,q_5$	$0,L,q_6$
q_6	B,R,q_1	$0,L,q_6$	$1,L,q_6$
q_7	Halt	B,L,q_7	Error

表1中标有Error的地方表示在计算中不会出现，这个图灵机程序所确定的计算可用众所周知的图2所示流程图来解释，其中$1*y$表示在字符串y的左面添符号1，$y*0$表示在y的右面添加一个0。

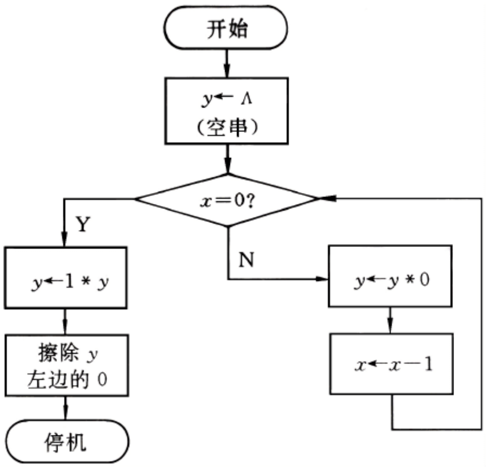


图2　计算$y=2^x$的图灵机流程图

可以用更确切的方式将图灵机的程序定义如下：以$Q=\{q_1,\cdots,q_m\}$表示有限状态集，$\Sigma=\{a_1,\cdots,a_n\}$表示磁带方格上的符号集，以$R,L,H$分别表示右移一格，左移一格或停机，一台确定型单带图灵机（的程序）由下述映射定义：

$$Q\times\Sigma\rightarrow\Sigma\times\{R,L,H\}\times Q$$

多带图灵机有一条输入带，k条工作带；每条带上有一个读写磁头与有限状态控制器连接，它根据当前状态以及输入带和工作带上当前被扫描的符号决定下一步应该转到什么状态，应在工作带和输出带上写下什么符号以及每个磁头各自的移动方向是什么。非确定型图灵机与确定型图灵机的区别在于：机器从一个状态转到下一个状态时不是唯一的，它可以在两个或更多个状态中择取。

如果将字母表Σ上的一个字$\alpha\in\Sigma^*$放在图灵机磁带上作为输入，图灵机M的磁头扫描在字α的开头字母上，并从状态q_1（定为初始状态）开始工作，则出现两种情况：要么在工作有限步后机器执行到某条停机指令停机，不再工作，这时磁带上输入的字α变成了输出的字β；要么机器永不停机，这时机器对输入的字α无输出。这样，一台图灵机M定义了一个从Σ^*到Σ^*的部分映射，称为图灵部分可计算映射，记为ψ_M；如果这个映射是处处有定义的，则称为图灵可计算映射。Σ^*的任一个子集合（语言）L称为递归可枚举的，若有一台图灵机M使$\mathrm{dom}\ M=L$；L称为是递归的（可判定的），若有一台图灵机M使得ψ_M恰为L的特征函数。判定问题是计算机科学理论研究中的一个重要问题。例如：已经证明，前后文有关文法的空虚性问题，Post对应问题等都是不可判定的。在计算机科学中，证明许多问题的不可判定性都是采用划归到图灵机的停机问题上。可以如下叙述图灵机的停机问题：不存在一种算法来判定：对任意图灵机输入任意字α计算是否停机。

表面看来，图灵机的计算功能似乎很弱。但只要提供足够的时间（允许计算到足够多的步数）以及足够多的空间（允许使用足够长的磁带），则其力量是非常强的，足以代替目前的任何计算机。图灵在设计了他的单带模型后提出：凡是可计算的函数都可以用一台图灵机来计算。这就是著名的图灵论题。A. Church在提出λ演算时也说，可计算函数都是λ可计算的，这就是有名的丘奇论题。而后证明这是两个等价的命题，以后称之为图灵-丘奇论题。论题中使用了直观的非数学的“可计算函数”一语，因而论题本身是不能用数学方法来论证的，只能为实践所检验。大半个世纪以来，数学家、计算机科学家提出了各种各样的计算模型都被证明是同图灵机器等价的。这一论题已被当作公理，它不仅是计算机科学的基础，也是数学的基础之一。

图灵机器的计算对时（计算步数）空（磁带长度）是不加限制的；然而，现时世界对时空的限制却是必不可少的，如果对一个长为1000的输入字要计算2^{1000}步，这在今日的大型计算机乃至可以想象的将来的计算机上都办不到。因此，自20世纪60年代起，对时空受限的图灵机器理论发展起来了，形成了今天的计算复杂性理论分支，出现了著名的尚未解决的所谓P与NP问题。P是这样的语言类（即Σ^*的子集L组成的类）：$L\in\mathrm{P}$当且仅当存在一台确定型的图灵机M和一个多项式f使得对任何输入字$\alpha\in\Sigma^*$，$\alpha\in L$当且仅当给图灵机M输入α时计算在$f(|\alpha|)$步内停机并处于接受态，这里$|\alpha|$是字α的长度；NP是这样的语言类：$L\in\mathrm{NP}$当且仅当存在一

台非确定型的图灵机 M 和一个多项式 f 使得对任何输入字 $\alpha\in\Sigma^*$，$\alpha\in L$ 当且仅当给图灵机 M 输入 α 时计算在 $f(|\alpha|)$ 步内停机。P = NP 吗？近年来已发表了大量论文探讨它；数学、计算机科学中的许多重要问题都同它有关，例如现代公钥密码体系就建筑在 P≠NP 假定上。

参考文献

1. Kleene S C. 元数学导论. 莫绍揆，译. 北京：科学出版社，1985

2. Rogers H. Theory of recursive functions and effective computability. New York：McGraw - Hill，1967

3. 李祥. 可计算性理论导引. 贵阳：贵州人民出版社，1986

（李祥）

tulun

图论（graph theory） 研究边和点的连接结构的数学理论。一个图 G 是一个三元组 $\langle V,E,\psi\rangle$，其中 V 是一个非空有限集，称为 G 的顶点集合，E 是有限集，称为 G 的边集合，ψ 是从边集 E 到顶点偶对集合的函数。图 G 的顶点数目称为它的**阶**。对任意的边 $e\in E$ 和顶点 $a,b\in V$，若 $\psi(e)=(a,b)$（无序偶对），则称 e 为从 a 到 b 的**无向边**；若 $\psi(e)=\langle a,b\rangle$（有序偶对），则称 e 为从 a 到 b 的**有向边**。这时，称 a 为 e 的起点，b 为 e 的终点，也称 e 关联于 a 和 b，顶点 a 和 b 邻接。图中不与任何顶点邻接的顶点称为孤立点。所有顶点全为孤立点的图称为零图。关联于同一顶点的两条边称为邻接边，关联于同一顶点的边称为自圈。对任意的边 $e_1,e_2\in E$，若 $\psi(e_1)=\psi(e_2)$，则称 e_1 和 e_2 为平行边。不含自圈和平行边的图称为**简单图**，有平行边的图称为多重图，每一条边均为有向边的图称为有向图，每一条边均为无向边的图称为无向图。既有无向边又有有向边的图称为混合图。若把有向图中每条有向边改为无向边，则称所得到的无向图为该有向图的**底图**。**路径**是一个边的有穷序列 $e_1,e_2,\cdots,e_n(n\geqslant 0)$，其中 $e_i(1\leqslant i<n)$ 的终点与 e_{i+1} 的起点相同。若 e_1 的起点和 e_n 的终点重合，则称该路径为闭的，否则称为开的。如果闭路径通过各顶点不超过一次，则称它为**回路**。若从顶点 a 到 b 存在路径，则称从 a 可达 b，否则称从 a 不可达 b。在无向图 G 中，若任意两个顶点是可达的，则称 G 为**连通图**。

设图 $G=\langle V,E,\psi\rangle$ 和 $G'=\langle V',E',\psi'\rangle$ 同为无向图或同为有向图。若 $V'\subseteq V,E'\subseteq E$ 且 $\psi'\subseteq\psi$，则称 G' 为 G 的**子图**，记为 $G'\subseteq G$。若 $G'\subseteq G$ 且 $G'\neq G$，则称 G' 为 G 的真子图。若 $G'\subseteq G$ 且 $V'=V$，则称 G' 为 G 的生成子图。

设图 $G=\langle V,E,\psi\rangle$ 且函数 $w:E\to\boldsymbol{R}_+$（$\boldsymbol{R}_+$ 为非负实数集合），则称 $\langle G,w\rangle$ 为**加权图**，w 为权函数，$w(e)(e\in E)$ 为 e 的权。如果 G 为有向图且其底图为连通的，则称加权图 $\langle G,w\rangle$ 为网络，w 为容量函数，$w(e)(e\in E)$ 为 e 的容量。

E. Euler 在 1936 年解决了著名的哥尼斯堡七桥问题，从而成了图论的创始人。七座桥将流经东普鲁士的哥尼斯堡市之普莱格尔河中的两个小岛及岛与河岸连接起来（见图 1）。问题是要从四块陆地 A,B,C 和 D 中任意一块出发，经过每座桥恰好一次，再回到起点。人们经过多次试验都没有成功，这就是著名的七桥难题。最后 E. Euler 指出，七桥难题是无解的。他将陆地用顶点代替，桥用边代替，从而把七桥难题转化为图 2 所示的欧拉回路问题。

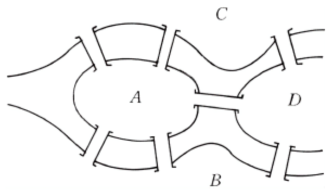


图 1　哥尼斯堡七桥示意图

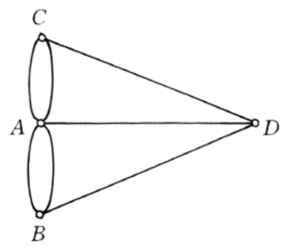


图 2　哥尼斯堡七桥模拟图

图论产生后，随即经历了一段半停滞时期。到了 19 世纪中叶，才又活跃起来，并在应用于化学和电路分析等方面，取得了重要成果。接着半个多世纪又几乎陷于停顿，直到 20 世纪中期，图论及其应用的研究才得到迅速发展，近年来尤为显著。

到目前为止，图论所研究的基本内容有：图及其表示，**有向图**、**无向图**和多重图，**树**，**平面图**，二部图和网络等。图论所研究的典型问题有：

最短路径问题 它有两种类型，其一是求从加权图中任一给定顶点到其余各个顶点的最短路径，其二是求加权图中任意两给定顶点间的最短路径。这里的“最短”是指路径的加权长度最短。

着色问题 在对一个图的顶点进行着色时，为了保证相邻接顶点的着色均不相同，至少需要多少种颜色？已经证明 5 种颜色是足够的，3 种颜色是不行的。人们猜测 4 种颜色最合适，这就是著名的四色问题。

环球旅行问题 威廉·哈密顿 1957 年发明了一种游戏，就是在实心正十二面体的 20 个顶点处标以有名的城市名字，要求游戏者找一条沿着各边通过每个顶点恰好一次的闭路径（称为回路），即“环球旅行”。由此引出了哈密顿回路和哈密顿图等概念。迄今为止，尚未获得图中存在哈密顿回路的充要条件。

欧拉路径问题 通过一个图中每条边恰好一次的路径称为该图的欧拉路径，闭欧拉路径称为欧拉回路。一个图中是否存在欧拉路径，以及如何求出存在的欧拉路径，就是所谓的欧拉路径问题。它已由 E. Euler 彻底解决了。

平面图问题 在现实生活中，常常要画一些图形，并希望边与边之间尽量减少相交的情况，例如印刷线路板上的布线、交通道路的设计等。那么，一个图能否画在一个平面上，使其边不在非顶点处相交叉呢？由此引出了平面图、对偶图等概念。

极小连通问题 有 n 个城市 $A_1,A_2,\cdots,A_n$，在这些城市之间要建造一个公路系统，使得旅行者从任何一个城市可以到达其他所有城市。那么，最少要修建多少条公路，才能构成上述公路系统？怎样使总的费用达到最小？由此引出了树、生成树、最小生成树等概念。

完美匹配问题 有 n 个人 $x_1,x_2,\cdots,x_n$ 和 m 台机器 $y_1,y_2,\cdots,y_m$。若 $x_i(1\leqslant i\leqslant n)$ 可以操作机器 $y_j(1\leqslant j\leqslant m)$，就在 x_i 和 y_j 之间连一条边。在工作过程中，每个人只能操作一台机器，且每台机器也只能由一个人操作。试问：应如何适当地安排，才能使尽可能多的人工作。由此引出了二部图、极大匹配、完美匹配等概念。

图的矩阵表示问题 为了有效地利用计算机对图进行处理，首先，必须给出一个图在机器内部的存储形式。为此，引出了图的邻接矩阵、可达矩阵、关联矩阵等。通过图的矩阵表示，也使我们能用代数工具研究图的一些性质。

网络流问题 假设把某种物资从产地 A 运到销售地 B，途经若干中转站。每两个中转站之间的货运量都有一个最大限量（称为该段道路的容量），如何制定从 A 到 B 之运输量最大的方案，就是所谓网络的最大流问题。

图论是一门应用性很强的学科。应用图论来解决运筹学、化学、生物学、网络理论、信息论、控制论、博弈论和计算机科学的问题，已显示出极大的优越性。

（张强）

tulun suanfa

图论算法（graph algorithm） 关于**图论**问题的求解算法。图是计算机科学中一种常见的数据结构，因而与图有关的算法在计算机科学中也十分重要。采用图来描述的有意义的计算问题成百上千。

一般来说，图在计算机中的表示方法有邻接矩阵表示法、二进制位向量表示法、邻接表表示法、邻接向量表示法和关联矩阵表示法。对于一些特殊的图，还有一些特殊的表示方法。总的来说，各种表示方法各有其优缺点，采用不同的表示方法，可获得不同的时空性能。对于具体的问题，应根据对图要进行哪些操作、按什么方式操作以及各种操作的使用频率等因素来决定采用何种表示方法，以获得时空性能令人满意的算法。

搜索一个图是指从某点出发按某种方式沿着该图的边前进，以访问该图的所有顶点。图的搜索问题是图论中的基本问题，许多图论算法以图的搜索算法为基础。宽度优先搜索和深度优先搜索是两种基本的搜索图的算法。

给定图 $G=(V,E)$ 和源 $s\in V$，*宽度优先搜索*只有在发现了与 s 相距 k 的所有顶点之后，才开始发现任何与 s 相距 $k+1$ 的顶点，这里两点之间的距离是两点之间“最短路径”的长度，即其中包含的边数。*深度优先搜索*遵循的策略是尽可能在图中搜索更深的顶点。其中，对最近发现的顶点 v，若有尚未考察的边离开 v，则考察之。在考察完所有这些边后，搜索将回溯去考察离开顶点 u 的边，这里顶点 v 是从顶点 u 发现的。深度优先搜索可用于形成有向无圈图的拓扑序，还可用于将有向图分解为强连通分支。

在最短路径问题中，给定一个加权的有向图 $G=(V,E)$，其权函数 $w: E\to\mathbf{R}$ 把边映射成实数权值。路径 $p=\langle v_0,v_1,\cdots,v_k\rangle$ 的权是其中各边的权之和 $w(p)=\sum_{i=1}^{k}w(v_{i-1},v_i)$。从 u 到 v 的最短路径权 $\delta(u,v)$ 定义为：若存在从 u 到 v 的路径，则它是从 u 到 v 的所有路径的权的最小值，否则它为 ∞。从顶点 u 到顶点 v 的最短路径定义为具有权 $w(p)=\delta(u,v)$ 的任何路径 p。

在单源最短路径问题中，给定一个图 $G=(V,E)$，希望找出从某个给定源 $s\in V$ 到每个顶点 $v\in V$ 的最短路径。许多其他问题可用单源问题的算法来求解。单源最短路径问题求解算法中使用的主要技术是松弛，这种技术反复地减少每个顶点的实际最短路径权的上界，直到该上界等于最短路径权时为止。各种求解算法之间的差别在于它们松弛每条边的次数以及松弛边的顺序。在 Dijkstra 算法和有向无圈图的最短路径算法中，每条边刚好松弛一次。而在 Bellman-Ford 算法中，每条边要松弛多次。Bellman-Ford 算法虽然比 Dijkstra 算法更通用，但执行时间却比后者多。按照加权有向无圈图中顶点的拓扑序来松弛离开各顶点的所有边，可快速地计算单源最短路径。单源最短路径问题求解算法可用于求解差分约束系统。

在每对顶点间最短路径问题中，给定一个图 $G=(V,E)$，希望找出每对顶点 $u,v\in V$ 之间的最短路径。虽然可依次以每个顶点作为源，反复运行单源最短路径算法 $|V|$ 次来求解这个问题，但是还有一些更好的求解算法。一般来说，可采用动态规划的方法来求解这个问题。性能较好的求解每对顶点间最短路径问题的 Floyd-Warshall 算法也采用了动态规划的方法。Floyd-Warshall 算法的思想也可用于求有向图的传递闭包，在支持位向量运算的计算机上，该算法既省空间又省时间。求解每对顶点间最短路径问题的 Johnson 算法对于边较少的所谓稀疏图，其性能较好。该算法采用重置权技术，把原问题变换成另一个等价的问题来求解。

事实上，称为“闭半环”的代数结构可为求解有向图的路径问题产生一种一般的框架。许多算法都可看成闭半环上用于计算每对顶点间路径信息的一般算法的实例。

有关计算一个图的最小权生成树的算法参见**最小生成树**。有关计算一个有向图的最大流的算法参见**最大流**。

参考文献

Cormen T H, Leiserson C E, Rivest R L. Introduction to algorithms. Cambridge, MA: The MIT Press, 1990

（殷建平　陈火旺）

tu wajue

图挖掘(graph mining)　对图或网络表示的数据的挖掘。是**数据挖掘**的分支，相关算法已被广泛应用于信息检索、图像处理、生物信息学、社会网络分析、化学分析、电路设计等领域。

图挖掘早期的主要任务是频繁子图挖掘，旨在识别图中频繁出现的子图模式。1994 年 Diane Cook 等提出的 Subdue 算法和 Kenichi Yoshida 等提出的 GBI 算法是该领域的最早工作。2000 年 Akihiro Inokuchi提出的 AGM 算法和 2002 年 Jiawei Han 等提出的 gSpan 算法是最具代表性的工作。AGM 基于**频繁模式挖掘**算法 Apriori 的思想，采用生成与测试方法挖掘给定图数据库中满足最小支持度和最小置信度的频繁子图。gSpan 基于频繁模式增长的思想，借助频繁模式树表示和挖掘给定图数据库中的频繁子图。

随着应用的拓展，图挖掘的主要任务扩展为：基于链接的排名，基于链接的分类，社区挖掘和链接预测等。

基于链接的排名旨在根据图或网络的链接结构计算节点的重要性，进而对节点进行排序。Web 的超链接结构与页面包含的主题紧密相关，从超链接结构可计算出网页主题与用户查询的相关程度，使得基于链接的网页排名算法成为当前主流搜索引擎的核心技术。Larry Page 和 Sergey Brin 提出的 PageRank 算法和 Jon Kleinberg 提出的 HITS 算法是两个最成功的网页排名算法，分别被应用于 Google 和 Altavista 等搜索引擎中。

基于链接的分类主要包括协作分类和半监督分类。协作分类根据链接关系将网络中节点赋予某一类标签。目前常用的协作分类算法包括 Gibbs 抽样、松弛标记、迭代分类和循环传播等算法。半监督分类根据部分训练样本用正或负标签标记图中所有节点，主要方法包括 Ross King 等首先采用的归纳逻辑程序法和 Thomas Gartner 等最先采用的核方法。

2002 年 Michelle Girvan 和 Mark Newman 最早发现现实世界网络中普遍存在着社区结构(相同社区中链接紧密，不同社区间链接稀疏)，并提了著名的社区挖掘 GN(Girvan-Newman)算法。社区挖掘旨在识别网络中的社区结构，主要的方法可分为基于优化的方法和启发式方法。前者将社区挖掘转化为优化问题，通过最优化预定义的目标函数挖掘社区结构，如谱图分割算法将社区挖掘转化为二次型优化问题，通过计算矩阵的特征向量来优化预定义的“截”函数。后者将社区挖掘转化为启发式规则的设计问题，如 GN 算法采用的启发式规则是：社区间链接的边界数应大于社区内链接的边界数。

链接预测是基于观测到的网络链接预测未知网络链接,可用于还原丢失链接和剔除噪声链接。2002 年 Lise Getoor 等提出的基于概率关系模型的链接预测方法是该领域早期的代表工作。主要的链接预测方法可分为:①基于相似度的方法是根据网络结构定义节点对的相似度,相似度大的节点对具有较大的链接概率;②统计关系模型方法是基于概率关系模型、关系马尔可夫网模型和逻辑回归模型预测节点对的链接概率;③基于随机网络的方法是将网络视为由多个随机子网络构成的随机网络模型,网络链接遵循二重伯努利分布,采用极大似然等方法估计出节点对的链接概率。

尽管已存在多种方法,图挖掘还面临一些未解决的问题:上述方法大都只能完成单一图挖掘任务,而实际复杂问题常常要求多个挖掘任务的有机集成;上述方法还不能有效处理包含多种类型节点和多种链接关系的异构网络;不能有效处理大规模网络和动态网络。

参考文献

1. Chakrabarti D, Faloutsos C. Graph mining: Laws, generators, and algorithms. ACM Computing Surveys, 2006, 38(1), 1-69

2. Aggarwal C C, Wang H X. Managing and mining graph data. New York: Springer, 2010

3. Yu P S, Han J, Faloutsos C. Link mining: Models, algorithms, and applications. New York: Springer, 2010

（杨博　刘大有）

tuxiang bianjie biaoshi

图像边界表示(image boundary representation)　把图像中的物体轮廓或边界用合适的数据结构或数字串表示的方法。图像边界表示的方法可分为链码表示、折线近似表示、ψ-S 曲线表示以及傅里叶描述子表示等。

链码表示　图像的边界是由一串离散的像素点组成的,如果将图像用一网格覆盖,使得像素点位于网格的交点上,那么边界可以看成是由一系列短线段组成的链,其中每个短线段正好是网格相邻交点的连线。

链码表示是将相邻短线段的方向向量进行编码。有 4 链码和 8 链码之分。8 链码有 8 个方向,分别用 0,1,2,3,4,5,6,7 共 8 个数分别表示 0°,45°,90°,135°,180°,225°,270°和 315°的 8 个方向(见图 1)。链码就是从起始点开始,沿曲线观察每一线段的走向,并用相应的方向码来表示,例如在图 2 中的曲线之链码为:0 1 2 2 2 3 2 2 1 0 0 0 0 0 7 6 5 5 5 6 7 1 1。

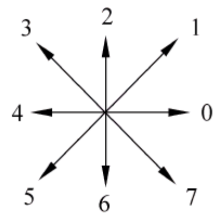


图 1　8 链码的 8 个方向

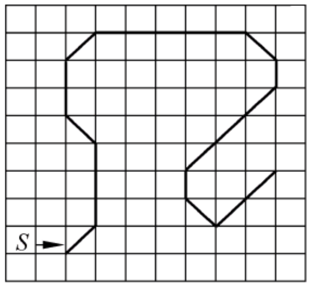


图 2　用链码表示曲线

为了确定链码所表示的曲线在图像中的位置,并由链码准确地重建曲线,必须标出起始点的坐标。但有时并不关心起始点的具体位置,特别是当曲线为闭合时,因为起始点位置的变化只引起链码位移的缘故。此时,可以通过改变起始点位置使表示起始点坐标的整数最小,这个过程称为链码的规格化。

折线近似表示　用许多折线段的组合来近似表示曲线边界的方法。其基本做法是将边界不断地用折线段划分下去,直到满足某一结束条件为止。具体方法如下:

(1) 将边界两端点用直线连接起来。

(2) 计算边界上的各点到该直线的距离,若其中最大距离小于给定阈值,结束。

(3) 将最大距离所对应的边界点作为划分点。

(4) 去掉连接端点的直线,将两端点分别与划分点用直线连接,转至(1)。

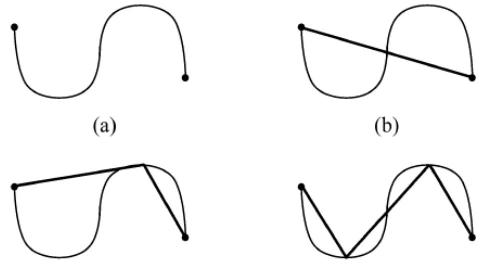
上述过程可用图 3 表示。

ψ-S 曲线表示　这种方法是通过测量边界上各点处的切线与 x 轴的夹角 ψ 和相对于某一起始点各切点的弧长 S 来表示边界的形状。

傅里叶描述子表示　这是一种闭合边界的常用表示方法。

任何闭合边界可以在 xy 复平面上表示为

$$u(n) = x(n) + jy(n) \quad n = 0,1,\cdots,N-1$$



(a) (b) (c) (d)

图 3　曲线的折线近似表示

这是一个以 N 为周期的周期函数,其傅里叶变换表示为

$$u(n) = \frac{1}{N}\sum_{n=0}^{N-1}\alpha(k)\exp\left(\frac{j2\pi kn}{N}\right)$$

$$0 \leqslant n \leqslant N-1$$

其中复系数 $\alpha(k)$ 为

$$\alpha(k) = \frac{1}{N}\sum_{k=0}^{N-1}u(n)\exp\left(\frac{-j2\pi kn}{N}\right)$$

$$0 \leqslant k \leqslant N-1$$

这里称 $\alpha(k)$ 为边界的傅里叶描述子。它具有如下特点:①边界的许多几何变形,如平移、尺度变化、旋转等都可以通过傅里叶描述子的简单运算得到;②用傅里叶描述子可以重构原边界;③利用傅里叶描述子可以将不同大小、不同朝向的类似形状加以匹配。

除上述几种的方法之外,还有二次、三次曲线等边界的表示方法。

参考文献

1. 赵荣椿,等. 数字图像处理导论. 2 版. 西安:西北工业大学出版社,1995

2. Jain A K. Fundementals of Digital Image Processing. Prentice-Hall Inc. ,1989

3. Gonzales R, Woods R. Digital Image Processing. Addison-Wesley Publishing Company Inc. 1992

(曾建超)

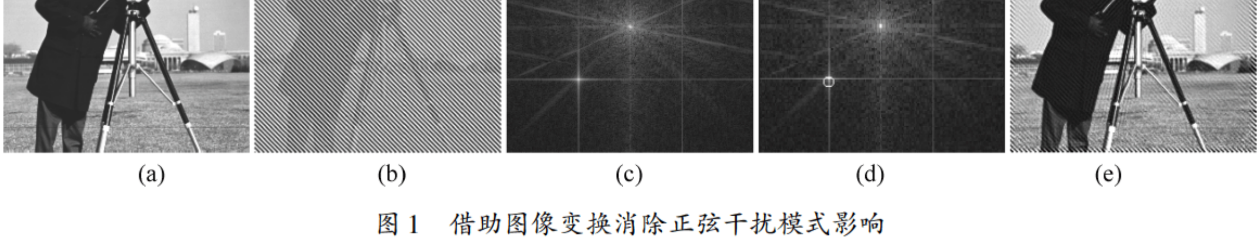
tuxiang bianhuan

图像变换(image transforms)　将图像以某种形式规则地从原有表达空间转换到另一个新表达空间的方法或过程。目的是利用新空间的特有性质对图像进行快速和有效的处理,例如频域滤波、干扰消除、数据压缩、特征提取等。下图为借助图像变换消除图像所受正弦干扰模式影响的一个实例。其中,图 1(a)为一幅原始正常图像,图 1(b)是图 1(a)受到周期性正弦干扰模式覆盖后的图像,图 1(c)是对图 1(b)进行傅里叶变换(见表 1)得到的新空间里的结果,图 1(d)表示将两个带阻滤波器(白圆圈)放置在图 1(c)中一对较明显的脉冲处,以把脉冲滤除掉,图 1(e)所示为对图 1(d)再取傅里叶反变换回到图像空间得到的结果,这里正弦干扰模式基本上就被消除了。

由上图可见,将图像转换到新空间并进行处理后,还要将处理结果转换回原空间以得到所需的加工效果。由上述步骤可知图像变换是双向的。一般正向变换简称变换而反向变换简称反变换。实现图像变换的方法有数字和光学两种形式,它们分别对应离散运算和连续函数运算。

如果图像变换可表示成级数展开的形式,则展开的基本函数也就是变换的核。对应正变换和反变换,有正变换核和反变换核。如果一个二维的变换核可分解成两个一维的变换核,则称变换是可分离的。如果这两个变换核具有相同的形式,则称变换是对称的。在计算具有可分离变换核的二维图像变换时,可将其分解成两个步骤,每个步骤计算一个一维变换。如果变换核是可分离的和对称的函数,则图像变换可表达成矩阵形式。如果变换矩阵为实矩阵,且变换矩阵的转置等于变换矩阵的逆,则称变换矩阵为正交矩阵(其中的任意两行或两列的内积为 0),而相应的变换称为正交变换。

图像变换的种类有很多,一些常用的如表 1 所列。



(a) (b) (c) (d) (e)

图 1　借助图像变换消除正弦干扰模式影响

表 1　一些常用图像变换简介

名称	定义和简介
短时傅里叶变换	一种特殊的傅里叶变换。"短时"一词源于一维信号处理领域,代表有限时间。而对时间的限定是靠加窗函数来实现的,所以也称加窗傅里叶变换。在二维图像处理中,与傅里叶变换需要知道图像在整个坐标平面上的分布才能计算单个频率上的频谱分量不同,短时傅里叶变换只需知道局部区间上的图像就可计算出单个频率上的频谱分量
傅里叶变换	一种得到广泛应用的可分离变换和正交变换。它能将满足一定条件的某个函数表示成三角函数(正弦或余弦函数)的线性组合,从而将图像从空间域变换到频率域。它的基本思想首先于 1807 年由法国数学家和物理学家傅里叶提出。 图像处理中一般考虑二维离散傅里叶变换。直接对一幅 $N \times N$ 的正方形图像进行二维傅里叶变换所需要的计算量为 $O(N^4)$,但根据傅里叶变换的分离性,可以将一个二维傅里叶变换转换为两个一维傅里叶变换,所需要的计算量为 $O(N^2)$。另外,傅里叶变换还有快速算法,称为快速傅里叶变换(FFT)。傅里叶变换快速算法与原始变换算法的计算量之比是 $N/\log_2 N$。 傅里叶变换适合于其性质随空间稳定不变的图像。在图像处理中,将图像经过傅里叶变换后,可以方便地设计滤波器进行各种低通、高通或带通滤波
盖伯(Gabor)变换	一种用高斯函数作为窗函数(时间域和频率域的窗函数均为高斯函数)而得到的短时傅里叶变换。它在 1946 年由匈牙利裔物理学家盖伯进行声波信号的时频分析时提出。盖伯给出了如何将一个信号分解成简单波形而表示的公式,其主要想法是使用一个包含局部时间的窗函数去找出傅里叶变换中的局部信息。 盖伯变换很容易从一维推广到二维。由于盖伯函数是复值函数,因此在运算过程中要分别计算其实部和虚部。实际中常使用两个成对的实盖伯滤波器,它们都对某个特定频率和方向有强响应。其中一个是对称的,另一个是反对称的。 盖伯函数用于在频域中的不同尺度、不同方向上提取相关的图像特征,非常适合表达和描述纹理图像,所以经常用在纹理表达、描述和识别中
哈尔(Haar)变换	一种可分离变换和对称变换。它可基于矩阵运算来实现,所用的变换矩阵称哈尔矩阵,是正交矩阵。 哈尔变换于 1909 年由匈牙利数学家哈尔提出。它的特点包括:计算速度快,只用加法和减法;频率只分低频(直流值)和高频(1 和 -1)。它可看作小波变换的一种特殊情况
哈夫(Hough)变换	一种利用图像空间中曲线或曲面的解析表达与表达参数间的对偶性在图像空间和参数空间之间进行的特殊变换。 它于 1962 年由美国人哈夫提出,并在美国申请了专利。它的主要优点是利用了图像全局特性,所以受噪声和边界间断的影响较小,比较鲁棒。 一般的哈夫变换可用来检测满足解析形式的各类曲线并把曲线上的点连接起来。而广义的哈夫变换通过利用表格来建立曲线或轮廓点与参考点间的关系,还可检测不易用解析式表达的曲线或目标轮廓。它在利用图像全局特性将目标边缘像素连接起来组成目标区域的封闭边界中、或在直接对图像里已知形状的目标进行检测中都得到了广泛应用
霍特林(Hotelling)变换	一种基于图像统计特性的、离散的图像变换。它也常被称为特征值变换或离散 KL 变换。它是主元分析的基础,也称主分量(PCA)变换。 霍特林变换可消除原始数据之间的相关性。它用于图像压缩、滤波和特征抽取时,在均方误差意义下是最优的。但它的运算较复杂且没有统一的快速算法
拉东(Radon)变换	一种用线积分来定义的图像变换。它于 1917 年由奥地利数学家拉东提出。它把平面图像变换为沿直线的投影,是从投影重建图像的基础,广泛应用于**图像重建**中

续表

名称	定义和简介
拉普拉斯变换	实变量函数和复变量函数间的一种函数变换。 拉普拉斯变换对应二阶导数运算,可用于锐化和提取图像中的边缘
离散余弦变换	一种可分离变换、正交变换和对称变换。 离散余弦变换相当于只使用实数的傅里叶变换,所以对离散余弦变换的计算可借助对离散傅里叶变换的实部计算来进行。 余弦函数是偶函数,所以离散余弦变换隐含 $2N$ 点的周期性。与隐含 N 点周期性的傅里叶变换不同,余弦变换可以减少在图像分块边界处的间断,这是它在图像压缩中,特别是 JPEG 标准中得到应用的重要原因之一
小波变换	一种在时间(空间)和频率上都具有局部性质的变换。从字面上说,"小波"指小的波形。再具体一些,"小"是指它具有衰减性;而"波"则是指它的波动性,小波具有其振幅正负相间且快速减弱的震荡形式。小波变换指用有限尺度的或快速衰减的震荡波形来表示图像。 小波变换在广义上指一系列适合不同应用的变换,包括多尺度小波变换,快速小波变换,离散小波变换,小波分解,小波包分解等。 在一维信号处理中,小波变换具有时间-频率都局部化的特点,而且时间窗函数的宽度与频率(变换域)窗函数的宽度的乘积是一个常数。小波变换的这个特性在二维图像处理中也很有用,例如用小波变换对图像低频分量分析时可加宽空间窗,减小频率窗;而用小波变换对图像高频分量分析时可加宽频率窗,减小空间窗。这种根据图像频率高、低而改变窗口宽、窄的能力称为变焦特性。这种特性是小波变换能够提供**图像多分辨率处理**的基础。 与仅在频域上局部的傅里叶变换相比,小波变换是空间(时间)和频率的局部变换,因而能有效地从图像中提取细节信息,从而被誉为"数学显微镜"。它在图像处理中除可用于构建多分辨率表达外,还可用于消除噪声、分割图像、提取特征、融合图像等
沃尔什(Walsh)变换和哈达玛(Hadamard)变换	它们都是变换核的值为 +1 或 -1 的可分离变换和正交变换。 沃尔什函数于 1923 年由美国数学家沃尔什提出。哈达玛矩阵是哈达玛变换的基础,由法国数学家哈达玛提出。 进行沃尔什-哈达玛变换只需要做加减法,运算复杂度低,所以曾经得到过广泛的应用。但与傅里叶变换相比,它们缺乏明确的物理意义和比较直观的解释。 沃尔什变换与哈达玛变换的系数都只取 1 和 -1,但在行和列的次序上两个变换不同,这也是当数据长度 $N=2^n$ 时,n 为整数时,两个变换唯一的不同点。因为在绝大多数应用中有 $N=2^n$ 成立,所以沃尔什变换和哈达玛变换常混合使用,沃尔什-哈达玛变换常用来指两者中的任一个
斜变换	一种正交变换,其性质包括:常数基矢量、基矢量幅值从最大到最小以常数单调下降、高能量集中、有迭代快速算法。它适合于表达灰度渐变的图像

参考文献

章毓晋. 图像工程. 3 版(合订本). 北京:清华大学出版社,2012

(章毓晋)

tuxiang bianhuan yunsuan

图像变换运算(image transform operation)

参见**图像变换**。

tuxiang biaoshi

图像表示(image representation) 将图像或图像中的某些部分(例如物体的轮廓)或某些区域用合适的数据结构或数学式子表示的方法。

一幅图像通过分割和特征提取之后,往往被分成一些不同特征的部分。为了利用计算机对图像进行更高层次的处理(例如,进行图像景物理解、物体识别以及对图像进行各种描述等),需要将这些特征的部分用合适的数据结构或数学式子表示出来。例如,通过边界提取处理之后(参见**图像分割**),图像中相互关联的边界被连接起来,它可以大致说明图像中物体的形状。但这只是在像素层

次上的说明,是通过人来理解的,而计算机本身还并不容易判别出物体的确切形状。在轮廓提取的基础上,如果利用链码、傅里叶算子等方法将轮廓明确地表示出来,那么计算机就可以方便地判别出物体形状,进而达到检测、识别景物中的物体或利用**计算机辅助设计**(CAD)技术来合成物体形状的目的。

图像表示的几种常用方法是:边界表示方法、区域表示方法、几何特征表示方法、矩表示方法以及骨架结构表示方法(参见**图像边界表示**、**图像区域表示**、**图像几何特征表示**、**图像矩表示**以及**图像骨架表示**)。　(曾建超)

tuxiang chongjian

图像重建(image reconstruction)　根据在物体外部测量得到的数据(例如投影、反射),依照一定的物理和数学关系反演、计算出物体内部的物理量分布,最后用图像显示器以图像形式显示出物体内部物理量的分布的技术。

1917 年,数学家 Radon 证明:已知所有入射角 Ω 的投影函数 $U(P,\Omega)$,可以恢复唯一的图像函数 $f(x,y)$。Radon 变换与 Radon 反变换是图像重建技术的基础。

1971 年 Hounsfielld 发明头颅计算机断层成像技术(computerized tomography,CT);后来从单一的头部检查发展到身体某个部位的检查;20 世纪 80 年代到 90 年代,扫描速度突破了亚秒;90 年代后,螺旋 CT 技术使横断 CT 向连续扫描的螺旋 CT 过渡,多层螺旋 CT 从 4/16/32/40 层到 64 层 CT 广泛的临床应用,大大拓展了 CT 的临床价值。

有以下两种常用的图像重建技术。

1. 利用射线重建图像

(1) 图像重建流程如图 1 所示,其设备主要由以下三部分组成。

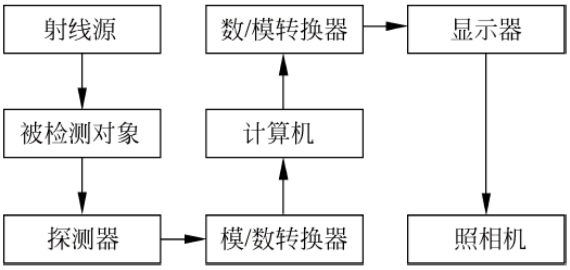


图 1　图像重建流程图

① 扫描部分:由射线源、探测器和扫描架组成;

② 计算机系统:将扫描收集到的信息数据进行运算、反演;

③ 图像显示系统:包括存储系统、显示器和照相机;

(2) 图像重建的算法

根据投影数据重建被测断面图像的方法通常有 5 种:联立方程法;反投影法;傅里叶变换法;卷积法;逐次逼近迭代法。

联立方程法是早期的方法,也是最基本的方法,但计算比较烦琐。

反投影法(或总和法)比较简单,但重建的图像质量较差。

傅里叶变换法是利用对象投影数据的傅里叶变换与对象图像的傅里叶变换之间的关系来进行重建的方法,这种方法已被广泛地应用于断层摄影技术中。

卷积法是由傅里叶变换法演变而来,这种方法的关键在于设计一个具有良好特性的重建滤波器。

逐次逼近迭代法的速度比解方程组法要快,比反投影法有较高的精度,所以也被广泛采用。近年来人们提出了不少提高迭代算法计算速度的方法,加上近年来计算机计算速度的迅速提高,迭代算法重新受到人们的关注。

由于应用的需要,局部重建算法(local reconstruction algorithm)也在近十年中有了较大的发展。在传统全局 CT 算法中,即使重建物体断面中一个小区域的图像,也得围绕整个断面采集投影数据。而局部重建算法,仅需围绕感兴趣区域及其邻域附近采集投影数据,即可重建感兴趣区域的图像。局部重建算法可减少数据采集时间和重建时间,降低人体(或生物体)的放射摄入量。

具体采用哪种算法,应根据:重建速度、空间分辨率、密度分辨率、对噪声的敏感度、重建对象特点,数据特点,以及应用需求等因素决定。

(3) 图像重建技术的应用

① 在医学中的应用(计算机断层成像):

以图像重建技术为基础发展起来的计算机断层成像技术最先在医学领域得到应用,推动了现代医学的发展。

CT 用 X 射线束对被检测人体某个部位的一个层面进行扫描,如图 2 所示。由于人体每个点对 X

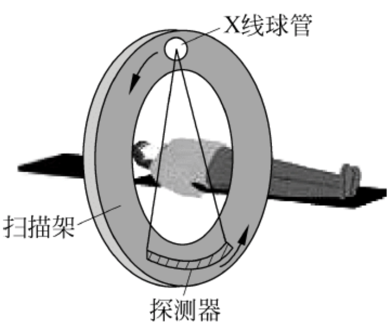


图 2　计算机断层成像

射线束衰减系数或吸收系数不同，投影数据因之不同。搜集每个角度下的投影，经反演、计算，可以得到该层面的图像，即 CT 图像。

多层螺旋 CT 技术，具有扫描速度快、时间分辨率高、安全、可靠、费用低廉、无创伤性等特点：可完成灌注成像、心脏成像、血管成像、虚拟内窥镜，可应用于外伤或急重症病人，可应用于尸检。

② 在工业中的应用：

工程 CT 技术（即工程层析成像技术），是将 CT 技术应用于工程所产生的。图 3 表示一个大型工程 CT 技术系统。工程 CT 可完成：缺陷检测、尺寸测量及装配结构分析、密度分布表征、场地和线路勘察与评价、工程地质灾害防治、农畜产品内部品质无损检测、农业无损检测、采油、找水等任务。特别是 CT 技术在安全检查中可以准确地发现并定位行李中隐藏的爆炸物。

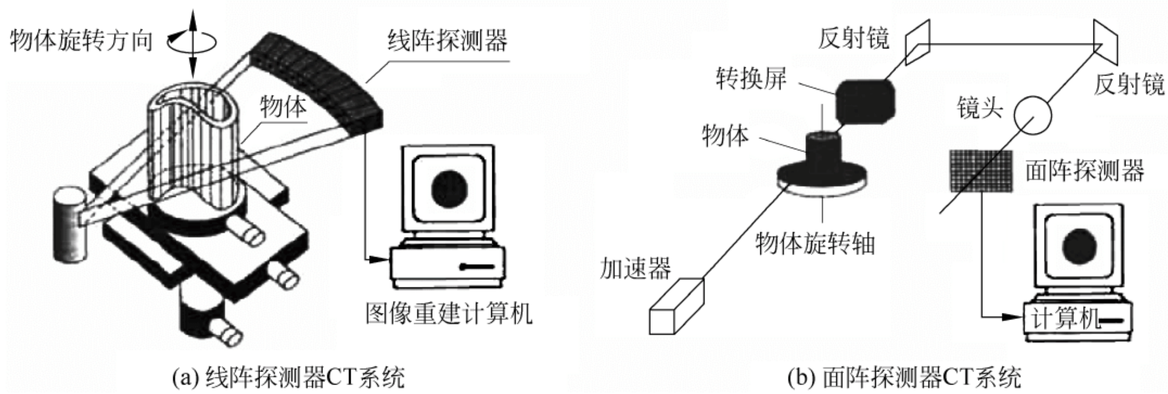


图 3　大型工程 CT 技术系统

③ 在军事上的应用：

制造与检测大型火箭发动机或小型精密铸件；检测炮弹的装药质量；检测火箭发动机推进剂的孔隙、杂质、裂纹以及推进剂、绝缘体、被套和壳体之间的结合情况；红外成像探雷技术。

2. 利用核磁共振原理重建图像

（1）核磁共振成像（nuclear magnetic resonance imaging，NMRI）原理

核磁共振成像的基本过程是：用磁场来标定人体共振核的空间位置，然后利用一定频率的电磁波，向处于磁场的人体照射，人体的各种不同组织的氢核，在电磁波作用下发生核磁共振，随之向外发射电磁波，测定这些能量信号并由计算机把这些信号加工成像。目前用于人体显像的装置大多是对氢原子核成像，这是因为氢元素在人体内普遍存在，且信号最强的缘故。

由于人体的各种不同组织的含氢量不同，且在同一组织中，正常和病变情况下，氢原子核的密度和弛豫时间存在着明显差异，因此，核磁共振就是把人体中的质子密度和弛豫时间的空间分布，用图像展现。如果使空间各点的磁场值均不相同，那么各处核的共振频率就不一样，而测出的 NMR 信号强度与质子的密集程度成正比，因而，如果把各点的核磁共振信号强度随频率分布显不出来，也就显不了共振核的空间分布，这就是核磁共振密度成像。

（2）核磁共振成像系统的组成（见图 4）

① 磁体系统：包括静磁场（一般采用超导磁体；）与梯度场两个部分。

② 射频系统：包括射频（RF）发生器；射频（RF）接收器两个部分。

③ 计算机图像重建系统：把由射频接收器送来的信号，根据与观察层面各体素的对应关系，经计算机处理，用不同的灰度等级显示出欲观察层面的图像。

CT 作为一种计算机层析成像技术，在医学、工业、地球物理、农业、工程检测和探测等多方面发挥

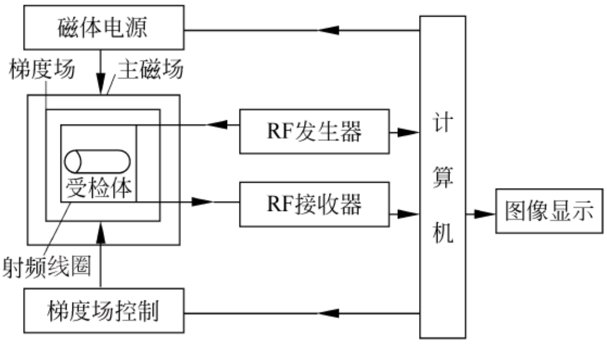


图 4　NMR 方框图

着越来越重要的作用,随着科学技术的进一步发展,CT 技术将向着多源、多排、多层方向发展,以求得扫描速度、覆盖范围、图像质量的同时改善。

近年来还出现了分子成像、μ 子成像等先进的成像技术。μ 子成像检测技术具有穿透力强、对高原子序数材料敏感、消耗能源少等特点,特别适合检测特殊核材料,是监控核材料走私的有效方式之一。相信在将来 CT 技术将会发挥越来越重要的作用。

参考文献

1. 刘东华,李显耀,孙朝晖. 核磁共振成像. 大学物理,1997,16(10)

2. 高丽娜,陈文革. CT 技术的应用发展及前景. CT 理论与应用研究,2009.3

（濮群）

tuxiang chuli de jiben yunsuan

图像处理的基本运算(basic operations in image processing)　指图像处理中常用的数学处理基本方法和算法,包括点运算、邻域运算、几何运算、变换运算以及形态学运算等。这些基本的运算是图像处理和分析中经常用到的对图像作预处理的方法,涵盖了数字图像处理的三大类数学工具(即正交变换,空域处理和形态学变换;参见**数字图像处理**)。图像携带了视觉(或其他)信息,但是由于条件限制、环境干扰或不同应用的特殊要求,图像“质量”往往不理想。为了有助于人或机器理解图像中的信息,常常有必要对图像作某种预处理。这些预处理包括改变图像的灰度分布、有意识地增强某些特征、或改变图像的空间分布、或对噪声进行过滤,以使图像更适合于人眼的观察、理解、或便于机器的分析。图像预处理的结果仍为图像,这些预处理在图像处理中往往称为图像增强(参见**图像增强**)。图像处理的基本运算中介绍的大部分技术都可以作为图像增强的基本技术。在图像分析和计算机视觉的系统中,这些预处理又被称为低层图像处理。从预处理方法的角度来看,有以下几种基本运算。

(1) 点运算　这种运算的处理结果 $g(x,y)$ 只和本像素点 (x,y) 的值 $f(x,y)$ 有关(参见**图像点运算**),即

$$g(x,y) = T_p[f(x,y)]$$

其中 T_p 表示灰度变换运算。或

$$g(x,y) = T_{mp}[f_1(x,y), f_2(x,y), \cdots]$$

其中 T_{mp} 为基本的算术或逻辑操作(+,-,×,/,AND,OR,XOR 等)。

(2) 邻域运算　是空间域滤波方法。像素点 (x,y) 的处理结果与 (x,y) 的邻域 S 内的像素点的值有关(参见**图像邻域运算**),即

$$g(x,y) = T_n[f(m,n)], \quad (m,n) \in S$$

S 为点 (x,y) 的邻域。

最常用的邻域运算是用卷积运算实现的线性滤波

$$g(x,y) = f(x,y) * h(x,y)$$

(3) 变换运算　将空间域图像 $f(x,y)$ 通过变换,例如傅里叶变换,变换为频域函数 $F(u,v)$,再进行滤波或处理。在处理后,通常可以再施行反变换回到空间域(参见**图像变换运算**)。

(4) 几何运算　几何运算实质上是图像的坐标变换(参见**图像几何运算**)。经过几何运算,像素 $f(x,y)$ 的坐标变为 (x',y')。几何运算的一般表达式为

$$x' = g_1(x,y), \quad y' = g_2(x,y)$$

(5) 彩色运算　包括伪彩色增强和彩色变换。伪彩色增强将灰度图像 $f(x,y)$ 映射为彩色图像 $[R(x,y), G(x,y), B(x,y)]$, $R(x,y) = F_r[f(x,y)]$, $G(x,y) = F_g[f(x,y)]$, $B(x,y) = F_b[f(x,y)]$,其中,F_r, F_g, F_b 为映射函数。

(6) 形态学运算　利用数学形态学的基本运算对图像进行处理,从而达到改善图像质量的目的。这是不同于空域滤波和频域滤波的一种新的图像滤波方法,称为形态域滤波方法(参见**图像形态学运算**)。一般的表达式为

$$X' = M(X,B)$$

其中,X 为处理对象,B 为结构元素,而处理结果是 X';M 表示形态学运算。

以上列举的基本运算不仅使用在**机器人视觉**、医学图像处理、遥感图像处理、文字识别等图像分析领域,而且广泛应用于影视特技、多媒体会议、**虚拟现实**等计算机应用领域中。这些基本运算可以单独

使用,也可以组合使用。例如,将点运算和几何运算结合,不仅改变了图像的灰度分布,也改变了图像的空间分布。在图像处理中应用广泛的小波变换可以看作是图像的邻域运算(如高斯平滑)、几何运算(图像比例变换)和图像正交变换(如余弦变换)的结合。

参考文献

1. Castleman K R. 数字图像处理. 朱志刚,等译. 北京:电子工业出版社,2002

2. Gonzalez R C,Woods R E. 数字图像处理. 2版. 阮秋琦,阮宇智,等译. 北京:电子工业出版社,2004

3. 许录平. 数字图像处理. 北京:科学出版社,2007

(朱志刚)

tuxiang de yasuo bianma

图像的压缩编码(compression and coding of images) 为提高图像的存储和传输效率,对静止或运动的数字图像进行数据压缩与编码的过程、方法和技术。

图像有静止图像和运动图像之分。前者指的是单幅图像,后者指的是内容随时间变化的一个图像序列,也就是日常所说的视频,通常每秒钟包含25~30 幅图像。

单幅图像的数据量可按下面的公式计算(以比特为单位):

图像数据量=图像水平分辨率×图像垂直分辨率×每个像素的比特数

为了节省存储数字图像时所需要的存储器容量,降低存储成本,特别是在互联网应用中,为了提高图像的传输速度,减少通信费用,大幅度压缩图像的数据量是非常必要的。由于数字图像中的数据相关性很强,或者说,数据的冗余度很大,因此对数字图像进行数据压缩也是完全可能的。而且,人眼的视觉有一定的局限性,即使压缩前后的图像有少许失真,只要限制在人眼无法察觉的误差范围之内,也是允许的。

图像数据压缩可分成两种类型,一种是无损压缩,另一种是有损压缩。无损压缩是指压缩以后的数据进行图像还原(也称为解压缩)时,重建的图像与原始图像完全相同。例如行程长度编码(RLC)、哈夫曼编码等。有损压缩是指通过解压缩还原图像时,重建的图像与原始图像有一些微小的误差,但不影响人们对图像的欣赏和对其含义的正确理解。变换编码、向量编码等都是有损压缩。评价一种压缩编码方法的优劣主要看 3 个方面:压缩比的大小,重建图像的质量(有损压缩时),以及压缩算法的复杂程度。

为了便于在不同的系统中交换图像数据,人们对计算机中使用的图像压缩编码方法制定了一些国际标准和工业标准。国际标准化组织(ISO)和国际电工委员会(IEC)联合制定了一组静止图像数据压缩编码的国际标准。如连续色调静止图像的编码标准 JPEG(ISO/IEC 10918)和 JPEG 2000(ISO/IEC 15444),二值(黑白)图像的编码标准 JBIG1(ISO/IEC 11544)和 JBIG2(ISO/IEC 14492)等。其中 JPEG 的适用范围较广,它能处理各种连续色调的彩色或灰度图像,算法复杂度适中,压缩比可控。对于内容为自然风光的彩色图像,压缩比为 10~20 时重建图像的误差一般不易为人眼所察觉。JPEG 标准已在数码相机和互联网图像存储与传输中广泛使用。

JPEG 2000 是基于小波变换的静止图像压缩编码标准,它的压缩比更高,图像质量也比 JPEG 好,既支持有损数据压缩也支持无损数据压缩,在图像品质要求比较高的医学图像分析和处理中已经得到比较广泛的应用。

除了 JPEG 和 JPEG 2000 标准之外,计算机和互联网中还广泛使用一些图像压缩编码的工业标准。表 1 是几种常用静止图像压缩编码标准的比较。

表 1 常用静止图像压缩编码标准

名称	压缩编码方法	类型	典型应用	开发公司(组织)
BMP	RLE(行程长度编码)	无损	Windows 应用程序	Microsoft
TIF	RLE,LZW(字典编码)	无损	扫描仪、桌面出版	Aldus,Microsoft
GIF	LZW	无损	互联网	CompuServe
JPEG	DCT(离散余弦变换),Huffman 编码	大多为有损	互联网,数码相机等	ISO/IEC
JPEG 2000	小波变换,算术编码	无损或有损	医学图像处理等	ISO/IEC

运动图像(视频)的数据量比静止图像大得多。例如1min的CCIR 601规格(演播室质量)的数字视频,其数据量约为1 GB,这样大的数据量无论是存储、传输还是处理,都是极大的负担,为此必须对数字视频进行大幅度的数据压缩。

数字视频的压缩编码方案很多,为了实现视频信息的可交换和互操作,必须制订一定的标准。数字视频压缩编码的国际标准由ITU、ISO和IEC分别或者联合制订,具体标准可参见**运动图像的压缩编码标准**。

参考文献

1. 国际标准化组织JPEG网站:http://www.jpeg.org/

2. 国际标准化组织MPEG网站:http://www.mpeg.org/

(张福炎)

tuxiang dian yunsuan

图像点运算(image point operation) 将图像各点的灰度逐点变换成另一灰度的运算。根据一幅图像进行的灰度变换,一般采用两种技术:一是根据事先定义的灰度映射关系进行变换的技术,二是根据图像直方图统计所进行的直方图修正技术。对于两幅或两幅以上图像对应点的点运算称为图像的代数运算(包括算术和逻辑运算)。

灰度变换 是通过函数关系 $g = T(f)$ 把给定的灰度级 $f \in [0,L]$ 映射到另一灰度级 $g \in [0,L']$ 的方法,常用的灰度变换有:

(1) 对比度扩展 当光照不足或不均匀,或者成像系统的动态范围小时,都会产生低对比度的图像。所谓"低对比度"的图像就是图像灰度集中在一个小范围内。当知道在某个灰度范围内的像素数目较多时,扩展该灰度范围可增强整个图像的视觉效果。下面是对比度扩展灰度变换的一个例子,其变换为

$$g = \begin{cases} \alpha f, & 0 \leqslant f < a \\ \beta(f-a) + g_a, & a \leqslant f < b \\ \gamma(f-b) + g_b, & b \leqslant f < L \end{cases} \qquad (1)$$

其中,α,β,γ 是直线斜率。如果需要扩展原图像的灰度区域 $[a,b]$,则斜率 β 应该大于1。

(2) 阈值化 阈值化常用于文字识别中的文字分割预处理。由于光照或传感器方面的噪声,输入的白纸黑字图像并不是黑白分明的。利用阈值化运算可得到黑白分明的二值化的图像。下面是利用阈值 t 进行二值化的灰度变换函数,即

$$g = \begin{cases} 0, & f < t \\ L, & f \geqslant t \end{cases} \qquad (2)$$

(3) 灰度反转 通过灰度反转可以得到输入图像的负像,灰度反转可用于医学图像处理、显示以及图像的打印等。灰度反转的变换函数可以是

$$g = L - f, \quad g,f \in [0,L] \qquad (3)$$

(4) 灰度窗口切分变换:将具有某一区间的灰度级的像素与其他部分("背景")分开,其表达式可以是

$$g = \begin{cases} L, & a \leqslant f \leqslant b \\ 0, & \text{其他} \end{cases} \quad \text{清除背景}$$
$$g = \begin{cases} L, & a \leqslant f \leqslant b \\ f, & \text{其他} \end{cases} \quad \text{保留背景} \qquad (4)$$

其中 $[a,b]$ 为所关心的灰度窗口。利用这种变换可检测出具有某一灰度区间内灰度级的所有像素,是图像灰度分析的常用方法。

(5) 灰度范围压缩 有时图像的灰度范围过大,以至于只有在某一灰度范围内的少量像素能够被分辨出来。灰度范围一般可以通过对数变换来进行压缩,例如

$$g = c\lg(1+f) \qquad (5)$$

其中 c 为一比例系数。对数变换压缩了灰度值高的范围,对灰度值低的范围反而有所增强。

直方图修正 灰度直方图表示了每种灰度在图像中出现的频率。灰度直方图修正技术是通过将直方图修改为所希望的形状以达到改善图像的目的。

(1) 直方图均衡化 ***直方图均衡化***就是把给定图像的直方图分布改变成均匀直方图分布。直观地讲,直方图均衡化导致了图像的对比度增加。需要注意的是,由于灰度等级的离散化,均衡化图像的直方图只是近似均匀分布。均衡化后图像的灰度范围扩大了,但其本质是扩大了一些量化层之间的间隔,而不是量化层的数目。有时,对整幅图像的直方图均衡化可能会抑制一些像素点不多但有用的细节。为解决这个问题,均衡化可逐块进行,所分的块可以交叠或不交叠。分块的方法尤其适用于不均匀的图像。

(2) 直方图规定化 均衡化的方法是直方图修正技术的一个特例,即希望修正后的图像灰度直方图为均匀分布。但在某些应用中,我们可能希望达到预先给定的分布,以便有意地强调某些灰度级的范围。这种方法称为***直方图规定化***。

图像代数运算 图像代数运算是两幅以上图像的运算。在过去一些图像系统中常用硬件来实现,但随着计算机计算速度的提高,图像代数运算软件

实现的速度也已不是问题。图像代数运算可用于图像平滑、运动检测、逐点校正和逐点逻辑判断等。

(1) 多图像平均与图像平滑　这种方法用于同一目标的图像可以重复拍摄的场合,目的是抑制图像摄取过程中由于种种原因引入的噪声。

(2) 图像相减与变化检测　在许多应用中,经常需要快速比较两个内容复杂的图像。一个简单的办法是将两幅对准的图像相减,这样它们之间的差别将会被强调出来。它可应用于医学诊断检测、安全监视,路面障碍物检测等。

(3) 图像相乘(除)与灰度校正　当图像灰度与实际景物亮度不匹配(或不一致)时,如果知道造成这种不一致是由于光照的不均匀或由于摄像机各点灵敏度的差异造成的,那么可以用逐点灰度校正,即把原图乘(或除)以一个校正系数图像的方法来处理。

参考文献

1. Castleman K R. 数字图像处理. 朱志刚,等译. 北京:电子工业出版社,2002

2. Gonzalez R C, Woods R E. 数字图像处理. 2版. 阮秋琦,阮宇智,等译. 北京:电子工业出版社,2004

3. 许录平. 数字图像处理. 北京:科学出版社,2007

(朱志刚)

tuxiang duo fenbianlü chuli

图像多分辨率处理(multi-resolution image processing)　对图像在多个分辨率上进行表达和处理的理论、技术和方法。它是基于在不止一个分辨率上对图像进行采集、表达、存储、传输、处理和分析等操作的多分辨率理论。该理论结合了多方面的技术和方法,包括**小波变换**的理论。因多个分辨率对应多个尺度,所以多分辨率理论也称多尺度理论。反过来,由粗及精对图像进行多尺度分析的方法也称为多分辨率分析。

实际中,图像中某种分辨率下不容易被获取的特性在其他一些分辨率下很容易被检测到。另外,通过使用多分辨率的技术常可以将一个复杂的函数分解为若干个简单的函数并对它们分别进行研究。所以利用多分辨率技术常可以更有效地提取图像特征,获取图像内容。典型的图像多分辨率处理包括图像多分辨率表达、图像多分辨率分解和重建、图像多分辨率变换等。

图像多分辨率表达

要在多分辨率下对图像进行处理,首先需对图像进行多分辨率表达,并建立各个分辨率空间的联系。金字塔是一种有效的多分辨率表达结构。借助金字塔对图像和目标进行多分辨率表达在本质上是对区域的分解,它将图像分解成多个层次,上下层间有"父子"关系,同层间有"兄弟"关系。

对同一幅图像采用不同的分辨率表达后,则在图像数据的表达中除了一般使用的空间分辨率外,又多了一个刻画当前分辨率层次的尺度参数。包含一系列有不同分辨率的图像的数据结构可被称为尺度空间(scale space)。

图像多分辨率分解和重建

多分辨率的操作要涉及对图像进行多分辨率分解和重建。小波是多分辨率图像分析的重要工具。在多分辨率小波图像分析中,采用缩放(尺度)函数来构成原始图像的一个近似序列,这个序列相邻两项之间的分辨率一般相差一个系数2。从离散的角度,多分辨率分解和重建也可通过采样和插值实现。

图像多分辨率变换

多分辨率小波变换是在多个分辨率上进行的小波变换。利用多分辨率小波变换的结果可以使对图像的小波分析聚焦到图像中的间断点、奇异点和边缘。例如,通过对多分辨率小波变换结果的非规则采样(irregular sampling)可获得小波模极大值,该值能表达图像中的边缘位置。

为获得多分辨率表达所采用的多分辨率变换技术可分成三大类,分别是尺度-空间技术(借助高斯滤波获取图像中的极值点),时间-尺度技术(借助小波变换自适应地获取图像中随分辨率变化的特性)和时间-频率技术(借助盖伯滤波获取图像的空间和频率局部特性)。

参考文献

章毓晋. 图像工程. 3版,合订本. 北京:清华大学出版社,2012

(章毓晋)

tuxiang fanxiang lübo fuyuan

图像反向滤波复原(image restoration by inverse filtering)　针对**图像退化**中的散焦退化现象的一种**图像复原**的过程和方法。对于一幅 $M \times M$ 点的数字图像 $g(x,y)$(原始图像)经成像光学系统作用后得退化图像 $f(x,y)$,数学上可以用 $g(x,y)$ 与光学系统点扩展函数 $h(x,y)$ 的卷积计算,即

$$f(x,y) = g(x,y) * h(x,y), \quad x,y = 0,1,\cdots,M-1$$

再加上噪声 $n(x,y)$ 得

$$f(x,y) = g(x,y) * h(x,y) + n(x,y),\quad x,y = 0,1,\cdots,M-1$$

据卷积定理,在频域中

$$F(u,v) = G(u,v)H(u,v) + N(u,v),\quad u,v = 0,1,\cdots,M-1$$

$$G(u,v) = \frac{F(u,v) - N(u,v)}{H(u,v)},\quad u,v = 0,1,\cdots,M-1$$

当噪声 $N(u,v)$ 小到可以忽略,即 $N(u,v) \approx 0$ 时

$$G(u,v) = F(u,v)/H(u,v),\quad u,v = 0,1,\cdots,M-1$$

对 $G(u,v)$ 进行傅里叶逆变换即可得到原始图像,即

$$g(x,y) = {}^{-1}[F(u,v)/H(u,v)],\quad x,y = 0,1,\cdots,M-1;\ u,v = 0,1,\cdots,M-1$$

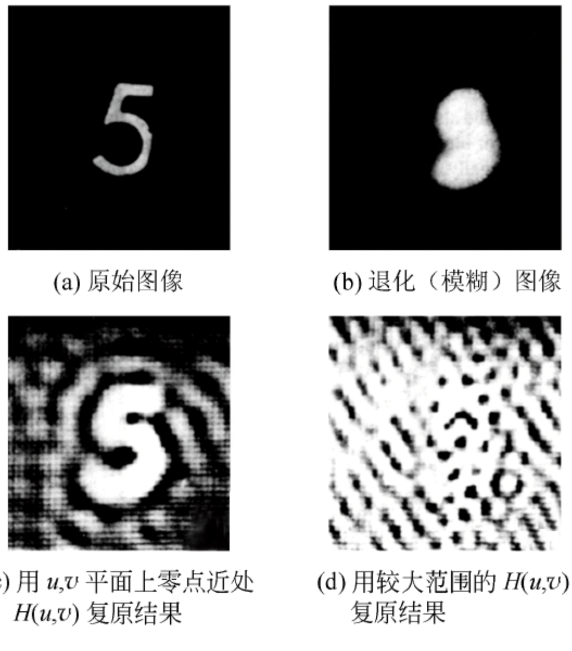
这种复原方法称为反向滤波法复原。

上面分析是在假设 $N(u,v)=0$ 的条件下,实际上 $N(u,v)$ 不为零。有噪声时,反向滤波得出的是原始图像的近似值 $\hat{g}(x,y)$。在频率域中,原始图像的近似值的傅里叶函数为

$$\hat{G}(u,v) = G(u,v) + N(u,v)/H(u,v)\quad u,v = 0,1,\cdots,M-1$$

注意,近似值 $\hat{G}(u,v)$ 与准确值 $G(u,v)$ 的差是 $N(u,v)/H(u,v)$。$\hat{G}(u,v)$ 经傅里叶逆变换得到 $\hat{g}(x,y)$。

图 1 给出一个用反向滤波法对一个含有噪声模糊图像复原的例子。



(a) 原始图像　(b) 退化(模糊)图像

(c) 用 u,v 平面上零点近处 $H(u,v)$ 复原结果　(d) 用较大范围的 $H(u,v)$ 复原结果

图 1　用反向滤波进行图像复原
(此图取自参考文献 3)

将图 1 中的(c)和(d)比较,显然(c)的效果比(d)好,因为 $H(u,v)$ 随着 u,v 频率变量增加而衰减,但 $N(u,v)$ 变化不大的特点,(c)只计算零点附近低频区的 $G(u,v)$,因为 $G(u,v)$ 和 $H(u,v)$ 在低频区的值都较大,所以 $N(u,v)/H(u,v) \ll G(u,v)$,复原效果较好。(d)在较大范围内取 $G(u,v)$,包括了 $H(u,v)$ 趋于零的值,而 $G(u,v)$ 已衰减,所以 $N(u,v)/H(u,v) \gg G(u,v)$。由于噪声淹没了图像,导致了复原失败。由此可见,用反向滤波进行图像复原不适合于信噪比小的图像。

参考文献

1. 贾永红. 数字图像处理. 2 版. 武汉:武汉大学出版社,2010

2. 赵荣椿,等. 数字图像处理导论. 2 版. 西安:西北工业大学出版社,1995

3. Rafael C, Gonzalez/Paul Wintz. Digital image processing. 2nd ed. Addison-wesley, 1987

4. Jain A K. Fundementals of digital image processing. Prentice-Hall Inc., 1989　(李树青)

tuxiang fenge

图像分割(image segmentation)　将图像分成特性相似的区域并提取出其中感兴趣目标区域(常对应场景中有意义的景物)的技术和过程。特性可以是灰度、颜色、纹理等视觉特性,也可以是其他本征特性或语义特性;相似可以是数值相等或相近,也可以是其他规律的接近;目标区域可以是连通的或不连通的。图像分割的结果是要把目标定位和提取出来,是一般意义上的目标检测。

至今为止,人们所提出的各类图像分割方法已有几千种,对它们的分类可考虑采用两个准则:①像素间的特征值性质;②计算策略。第一个准则考虑图像中相邻像素之间在特征值方面的特点,以灰度图像为例,区域内部的像素一般具有灰度相似性,而在区域之间边界上的像素一般具有灰度不连续性。所以分割技术可据此分为利用区域间特征值不连续性的基于边界的技术和利用区域内特征值相似性的基于区域的技术。第二个准则根据分割过程中处理策略的不同而将分割技术分为并行技术和串行技术两类。在并行类技术中,所有判断和决策都可独立地和同时地做出;而在串行类技术中,早期处理的结果可被其后的处理过程所利用,并充分考虑了全局性质。一般串行技术所需计算时间通常比并行技术要长,但抗噪声等能力通常也较强。两个准

则综合考虑,可将图像分割技术分成4类,见表1。

表1 图像分割技术分类表

分类	灰度不连续性	灰度相似性
并行技术	基于边界的并行分割方法(参见**基于边界的并行图像分割方法**)	基于区域的并行分割方法(参见基于区域的并行图像分割方法)
串行技术	基于边界的串行分割方法(参见**基于边界的串行图像分割方法**)	基于区域的串行分割方法(参见**基于区域的串行图像分割方法**)

参考文献

1. Zhang Y J. Image engineering: processing, analysis, and understanding. Cengage Learning, Singapore. 2009

2. 章毓晋. 图像工程(中册)——图像分析. 3版. 北京:清华大学出版社,2012 (章毓晋)

tuxiang fenxi

图像分析(image analysis) 对图像中感兴趣的目标进行检测和测量,以获得它们的客观信息,从而建立起对图像和目标完整描述的技术总和。

图像分析是图像工程(包括图像处理、图像分析、图像理解)的三个层次或三大分支之一,参见图1。图像处理着重强调在图像之间进行的加工变换以改善图像的视觉效果,或减少图像存储所需的空间,或图像传输所需的时间。如果说图像处理是一个从图像到图像的过程,则图像分析是一个从图像到数据的过程。这里数据可以是对目标特征测量的结果,或是基于测量的符号表示。它们描述了图像中目标的特点和性质。而图像理解的重点是在图像分析的基础上,进一步研究图像中各目标的性质和它们之间的相互联系,并通过对图像内容含义的理解得出对原来客观场景的解释,从而指导和规划行动。

图像分析的主要功能模块也可见图1。它包括对图像的分割提取,对目标的表达描述和对目标性质参数的测量分析。为完成这些工作,还需要使用和借助许多其他领域的理论、工具和技术。图像分析的工作围绕其操作对像(目标)进行,图像分割就是要从图像中分离出感兴趣的目标;对目标的表达是要有效地表示目标;对目标的描述是要稳定地描述目标性质;而测量分析则是希望精确地获得目标

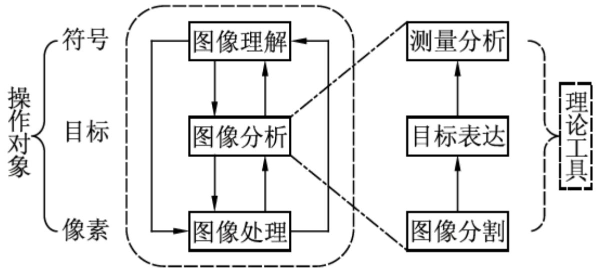


图1 图像分析及主要功能模块

的特性数据。由于目标是像素集合,所以图像分析对像素间的联系和关系(例如邻域,连通等)更为重视。在不同的应用中,人们对目标的分析要求也不同,所以图像分析又可分为颜色分析、纹理分析、形状分析、运动分析、目标空间关系分析等。

图像分析和模式识别(参见**模式识别**)有密切的关系。模式识别的目的是将不同的模式分类,将目标从背景中分割出来可看作将不同的区域区分出来,而为了进一步将区域分类,也需要确定描述区域特性的参数(特征)。这也是图像分析的目标。

早在20世纪60年代人们就研制了使用电视摄像机扫描图像进行分析的系统;70年代的图像分析系统已用软件替代了硬件;80年代和90年代图像分析系统得到了深入的研究、快速的发展和广泛的应用。近年来,系统的集成性进一步提高,片上系统(system on chip, SOC)也得到快速发展。这些硬件方面的进展不仅减少了成本,也促进了图像处理专用软件的发展。现在有许多图像分析系统和图像分析软件包已商品化。图像分析正向更快速、更精确、更自动、更智能的方向发展。

参考文献

1. 章毓晋. 图像工程(中册)——图像分析. 2版. 北京:清华大学出版社,2005

2. 章毓晋. 英汉图像工程辞典. 2版. 北京:清华大学出版社,2015 (章毓晋)

tuxiang fuyuan

图像复原(image restoration) 对一些退化的图像进行校正处理、滤除退化痕迹、恢复图像本来面目的技术。图像复原(或称图像恢复)应尽可能复原或逼近无退化的真实图像。

图像复原的质量主要取决于对图像退化过程的先验知识所掌握的精确程度。图像复原的一般过程是:弄清退化原因→建立退化模型→反向推演→恢复图像。

图像复原可以认为是**图像退化**的逆过程。

由于许多种退化都可以应用线性的位移不变模型来近似描述，这样就可把线性系统中的许多数学工具如线性代数用于求解图像复原问题，从而得到简洁的公式和快速的运算方法。

实际上，非线性和位移变的情况能更加准确而普遍地反映图像复原问题的本质，但难以求解。只有在要求很精确的情况下采用位移变的模型去求解，其求解也常以位移不变的解法为基础加以修改而成。

对于带有噪声的模糊图像 $f(x,y)$，要求复原出原图象 $g(x,y)$ 是不可能的。

复原方法有 4 种：**图像反向滤波复原**、**图像维纳滤波复原**、**图像运动模糊复原**和**图像修复**。

参考文献

1. 贾永红. 数字图像处理. 2 版. 武汉：武汉大学出版社，2010

2. 赵荣椿，等. 数字图像处理导论. 2 版. 西安：西北工业大学出版社，1995

3. Jain A K. Fundementals of digital image processing. Prentice-Hall Inc. , 1989

4. Rafael C, Gonzalez/Paul Wintz. Digital image processing. 2nd ed. Addison-Wesley, 1987 （李树青）

tuxiang gujia biaoshi

图像骨架表示（image skeleton representation） 利用图像骨架这种结构关系来描述图像中区域的方法。区域的中心骨架可以通过图像细化及中轴变换等方法来获取。

所谓中轴就是区域的对称轴线，中轴变换就是寻找对称轴线的过程。如果把轮廓线上的点称为轮廓点，那么骨架就是离两个或两个以上最近轮廓点距离最大的点之集合。

设一个区域为 R，其轮廓线为 B。对于 R 中的每个点 p，可找到其在 B 上的最近相邻点。如果 p 有多个这种相邻点，这些点即构成了 B 的中轴或骨架。显而易见，中轴的计算是与距离的定义直接相关的，常用的距离定义有欧几里得距离等。

为了确定骨架点，可以想象所考虑的区域是一片干草地，而区域边界的外面是池塘，当在区域的边界同时点上火以后，火就会烧进区域里面去。假如火的蔓延速度一样，从边界烧进来的火相遇的点就是骨架点。图 1 是 3 种不同的形状与它们的骨架。

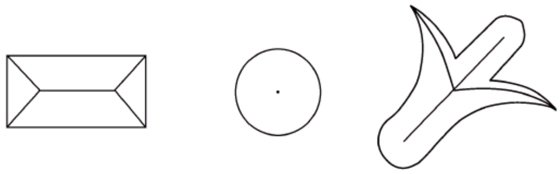


图 1　骨架

利用形态学运算也可以获得区域的结构特征及骨架（参见**图像形态学运算**）。

参考文献

1. 赵荣椿，等. 数字图像处理导论. 2 版. 西安：西北工业大学出版社，1995

2. Gonzales R, Woods R. Digital image processing. Addison-Wesley Publishing Company Inc. , 1992

（曾建超）

tuxiang huoqu

图像获取（image acquisition） 利用图像输入设备（例如摄像机）将图像输入计算机的过程和技术。常用的图像输入设备有摄像机、线阵摄像机、扫描仪等。传统的输入设备可以产生景物或图像的模拟信号，模拟信号经数字化后输入计算机（参见**图像表示**）。随着技术的进步，许多设备可以直接产生数字图像，这些设备中的图像传感器往往具有数字化功能。

模拟输入 尽管不同的图像采集设备的精度、速度及成本可能大不相同，但其原理却是相同的。在一个图像采集设备中，光学系统将景物或图片的“像”聚焦到成像面上。传感器将投射到其上面的光信号转换为电信号。传感器的输出部分将该电信号缓冲和放大，必要的话，还要转换为适于采集的电压（如全电视信号为 1V）。该电压经过**模数转换器**变成数字信号。时序和地址逻辑保证了输入模数转换器信号按适当的周期（时间间隔）采集，并存入图像存储器的正确位置。如果数字化的是彩色信号，则对于图像每个像素，传感器返回三个值，即红、绿、蓝分量。因此图像获取设备的基本部件为光学系统、传感器、模数转换器和图像存储器（帧存储器）。

电视摄像机 电视摄像机一般都装有一次能够扫描一整幅“图片”的电子线路。这种工作方式与广播标准和电视接收机标准密切相关，它更面向于人的观察方式而不是计算机图像处理所采用的方式。一整幅图像（我国采用的 PAL 制式中是 625 行扫描线，美国 NTSC 制式中是 525 行）为一帧，一帧

包含两场,即分别由一帧中奇数行扫描线和偶数行扫描线组成的奇数场和偶数场,这两场信号由摄像机电子线路顺序产生并传递。因此被传送的图像是交替的,奇数场先“画”在屏幕上,然后是偶数场。隔行扫描是为了防止图像闪烁。

扫描摄像机 扫描摄像机一次只获取一行图像。为了获取整幅图像,传感器必须用机械方法扫过所需视野,时间从几秒钟到数分钟,依所要求的信噪比而定。扫描摄像机不受电视标准的限制,能提供很宽的采集速度选择范围,并可获得高质量的图像。

线阵摄像机 线阵摄像机和扫描摄像机非常类似,但传感器和光学系统是固定在一起的,如果需要扫描,则由摄像机和物体的相对运动形成,一般情况下是物体移过摄像机的视野。线阵摄像机成像和产生数据的速度很高,常用于工业生产中。

文件扫描仪 文件扫描仪是扫描摄像机的另一应用。它可以获取静止的、平面的、大幅的高质量的图像,用于文字和图纸输入。

数字输入 随着数字图像获取技术的发展,直接获取数字形式数据的方法越来越普遍。其中有3个领域应用最广,即遥感成像、医学图像和距离成像。近来,在多媒体计算技术中也越来越普遍采用直接输出数字信号的摄像机。这些直接产生数字图像的数字照相机或摄像机有CCD和CMOS两种传感器,有些摄像机的成本很低,由于可以通过USB或其他标准接口直接输入计算机,已被广泛使用。

遥感成像 遥感成像包括航空和卫星成像。常用的卫星图像使用产生数字信号的传感器。

医学成像 医学成像用数字形式记录医学图像始于用X光构造人体内组织的断层图像(参见**图像重建**)。这种技术称为计算机断层摄影,简称CT。另一种是核磁共振成像(MRI),它利用原子核在强磁场下对电磁波的吸收产生图像。不论哪种方法都不直接测量图像数据,而是由传感器的输出计算出图像。

距离成像 距离成像的主要方法是雷达成像,其中包括超声雷达、激光雷达等。这种成像系统获取的数据是传感器到目标的距离而不是物体的亮度。距离是通过计算得到的,因此它产生数字形式的图像。

近年来全方位成像在视觉导航、视觉监视和视频消费领域得到不少应用,其特点是一次实时获取360°全方位的图像。使用鱼眼透镜、锥面反射镜或特殊设计的全景镜头都可以拍摄到360°全方位图像。

参考文献

1. Law A. Introductory computer vision and image processing. McGraw Hill Book Company, 1991

2. Yagi Y, Yachida M. Real-time omnidirectional image sensors. International Journal of Computer Vision-Special Issue on Omni-Directional Research in Japan, 2004, 58(3)

3. 朱志刚. 全视野时空视觉导航——真实景物的成像、建模与表示. 北京:高等教育出版社, 2001

(朱志刚)

tuxiang jihe tezheng biaoshi

图像几何特征表示(image geometric feature representation) 将图像中的物体或物体边界的几何特征用合适的数据结构或数学式表示的方法。这里的几何特征包括面积、紧凑性、偏心率、欧拉数、斜率密度函数等。

面积 设区域轮廓由$(x(s), y(s))$表示,其中s为弧长参数,则区域面积由格林公式计算。即

$$\text{面积} = \int_0^l \left(x\frac{\mathrm{d}y}{\mathrm{d}s} - y\frac{\mathrm{d}x}{\mathrm{d}s} \right)\mathrm{d}s \qquad (1)$$

其中l是区域的周长。

紧凑性 紧凑性=(周长)^2/面积

这是一个无量纲的量。显然,圆形区域的紧凑性为最小值,表明其紧凑性最好。

偏心率 定义为区域的最长弦A与其垂直方向上的最长弦B之比(参见图1),即

$$\text{偏心率} = A/B \qquad (2)$$

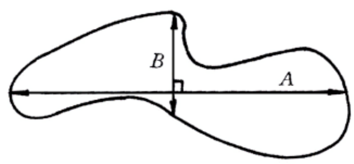


图1 偏心率

欧拉数 这是一个表示区域连通性的几何特征。它定义为

(连通域的个数)-(孔的个数)

斜率密度函数 虽然被称为斜率密度函数,实际上这是利用切线角密度来表示轮廓的方法。切线角密度可以用直方图来表示,横坐标表示轮廓切线的

角度,纵坐标表示具有某个切线角的切线数。显然,圆的斜率密度函数为一条水平直线,而折线轮廓斜率则会出现间断。利用斜率密度表示的例子见图 2。

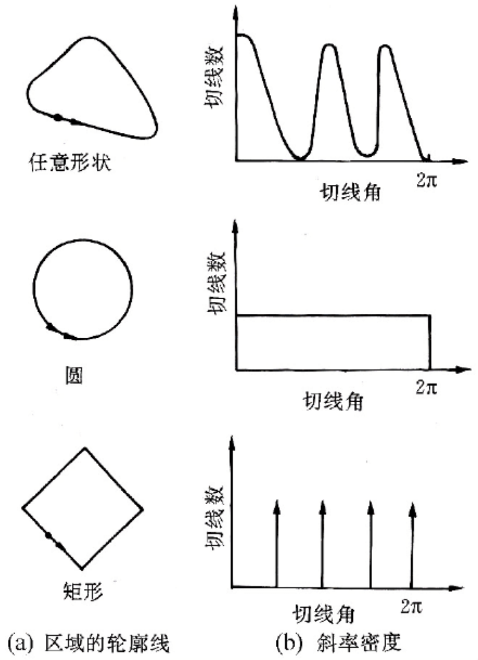


图 2　斜率密度函数表示

除了上述的几何特征外,还有对称性、形状数等一些有用的几何特征。

参考文献

Rosenfeld A, Kak A C. Digital picture processing. Academic Press, 1982

（曾建超）

tuxiang jihe yunsuan

图像几何运算(image geometric operation)

改变图像大小、形状、位置等几何属性的运算。图像几何运算包括图像重排序、图像缩放、剪贴、平移旋转和变形等。

图像重排序　通过改变图像像素的排序不仅可以产生一些有意义的平面或空间的显示效果,而且有时还有助于数据的处理或理解。例如,镜像变换可用于对称图像的比较,比较由光学成像引起的镜像(如胶片或投影片反置)以及医学中对称部位的图像(如左右肺)等。图像转置就是把图像的行、列交换,它在涉及向量及矩阵的几何运算中有用。例如,转置可用在图像的快速傅里叶变换中对行和列分别进行计算(参见**图像变换**)。图像旋转 $K\times90°$ ($K=1,2,3$)可以通过简单的像素重排序得到,除了显示效果外,90°旋转可用于不同方向排列的文字识别以及模板卷积的方向配合等。

图像缩放　对于数字图像,改变图像的大小就意味着改变每行、每列像素的个数,即改变分辨率。改变图像大小的一个最明显的目的是为了在特定比例的显示器上或打印设备上呈现图像。另外,不同分辨率图像在图像分析中也有重要的应用。放大图像时需要生成新的像素,一般可以利用邻域的像素值来估计新的像素值,这个操作称为插值。当缩小图像时,需要丢弃一些像素。这时,既可以简单地丢弃原来的一些像素,也可以考虑利用被丢弃的像素值来修改保留的像素值。一般来讲,要尽量保持原图中的形状、亮度及对比度。所以,在改变像素的数目时,可能要在效果和计算复杂度上取一个折中。

坐标映射和重采样　是改变图像尺寸的基本技术,其操作有两步:映射关系的确定和重新采样。当建立了原来像素坐标和结果像素坐标的映射关系后,可能产生具有带小数部分的坐标,这对于需要用整数坐标来表示的数字图像是个问题。因此,我们需要有第二个操作,即重采样。数字图像重采样和模拟数据采样有很大的不同。在数字图像重采样中,虽然我们知道需要的采样点的位置,但是往往不能直接获得新坐标下的像素的亮度,因为可能没有与此位置对应的原图像的采样值。因此,我们必须根据已知的邻点值利用插值来获取需要的采样值。常用的插值方法有:

(1) 最近邻插值　结果图像中每个像素的值选择与其在原图中对应点最接近的像素的值。

(2) 线性插值　尽管线性插值只是一种近似,在图像处理中由于其效果很好,故使用很多。

(3) 曲线插值　当需改变尺寸的图像噪声较大时,使用曲线插值更好。另一种方法是先进行图像平滑,再使用线性插值。

图像剪贴　在应用中经常需要取出图像的一部分单独进行处理,或者将两幅图像组合在一起等。这些操作分别称为裁剪和拼贴。裁剪操作是根据某种几何参数抽取出一块图像,即子图像;而拼贴操作是将几幅图像或子图像合成一个新图像。剪贴不仅用于图像处理,在图像显现上也很有用。在一般情况下,裁剪的图像区域不一定是矩形,可以是圆形、三角形、星形和其他任意形状。为了定义裁剪区域,一般有三种方法:

(1) 参数形式　这对于规则的形状较适用,例如矩形可用其左上角和右下角表示,圆可用圆心和

半径表示等。

(2) 区域边界表示　用一系列边界点来表示需要裁剪的区域,这在手工裁剪时常用。

(3) 标注方法　对于任意的裁剪区域,可以先用其最大外接矩形裁剪图像,然后对处于矩形区域内的像素加以标识,这样就形成一个标注图像,表示像素是否属于所关心区域。这种方法在图像生成和动画制作中有用。另外,由于绝大部分图像系统只能储存矩形图像,所以对于非矩形图像需要通过一定的填补方式表示成一个矩形阵列。

图像平移和旋转　平移、旋转可以校正输入图像的位置和方向,以便放正图像或进行图像比较。为了实现图像的平移和旋转,可以应用有关平移和旋转的变换矩阵。在图像平移或旋转时还要考虑两个问题:①变换前后图像的范围;②变换后新的图像点的灰度取值问题。第一个问题可通过裁剪解决,第二个问题可以通过插值方法解决。

图像变形　虽然平移、旋转可用于校正两幅或多幅图像之间的位置或方向,但有些图像在成像过程或数字化过程中的畸变并不能用二维的平移或旋转来改善。所以,往往需用更通用的图像变形方法。二维图像透视变换可以校正因摄像机纯旋转所产生的任意三维景物的图像变形或摄像机任意运动所产生的平面物体的图像变形。图像变形可用于分析、理解图像或表现实验结果等方面。

图像卷曲　是一种常用的图像变形技术。在前述的几种修改图像的方法中,包括图像缩放、平面旋转和透视变换等,都是基于相同的基本操作,即像素坐标的映射和图像的重采样来进行的。不同方法的区别只是映射的方式不同。另外,在前述的几种基本方法中,映射在几何意义上是精确定义而且和图像数据无关的。但是,更一般的复杂的图像映射有时不能够精确地定义,而且可能和图像本身有关,这种广义的几何变换称为图像卷曲,它可能既改变图像的大小又改变图像的形状。图像卷曲广泛地应用于遥感成像、视觉导航和虚拟现实等领域中。

参考文献

1. Castleman K R. 数字图像处理. 朱志刚,等译. 北京:电子工业出版社,2002

2. Gonzalez R C, Woods R E. 数字图像处理. 2版. 阮秋琦, 阮宇智,等译. 北京:电子工业出版社, 2004

3. 许录平. 数字图像处理. 北京:科学出版社, 2007

(朱志刚)

tuxiang ju biaoshi

图像矩表示(image moment representation)

把图像中的区域(物体的形状)用被称为"矩"的数学度量来表示的方法。由于图像区域的某些矩对于平移、旋转、尺度等几何变换具有一些不变的特性,所以矩的表示方法在物体(形状)分类、识别方面有重要意义。

设图像用有界函数 $f(x,y)$ 表示,其中 x,y 的定义域为 Ω,则 $f(x,y)$ 的 $(p+q)$ 阶矩为

$$m_{p,q} = \int\int_{\Omega} f(x,y)x^p y^q \mathrm{d}x\mathrm{d}y$$

其中 $p,q=0,1,2,3,\cdots$。

当 $p=q=0$ 时,$m_{0,0}$ 被称为图像 $f(x,y)$ 的零阶矩。它实际上是图像 $f(x,y)$ 灰度值的总和。对于二值图像来讲,如果灰度值是"1"表示物体,"0"表示背景,那么零阶矩 $m_{0,0}$ 就是图像中物体的面积。当 $p=1,q=0$ 时,$m_{1,0}$ 对二值图像来讲就是物体上所有点的 x 坐标的总和。类似地,$m_{0,1}$ 就是物体上所有点的 y 坐标的总和,所以

$$\bar{x} = \frac{m_{1,0}}{m_{0,0}},\ \bar{y} = \frac{m_{0,1}}{m_{0,0}}$$

就是二值图像中物体质心的坐标。

当 $p=2,q=0$ 时,$m_{2,0}$ 就相当于物体绕 y 轴旋转的转矩。类似地,$m_{0,2}$ 相当于物体绕 x 轴的转矩。

为了获得矩的不变特征,往往采用中心矩以及规一化的中心矩。中心矩定义为

$$m'_{p,q} = \int\int (x-\bar{x})^p (y-\bar{y})^q f(x,y)\mathrm{d}x\mathrm{d}y$$

这里,$\bar{x},\bar{y}$ 是物体重心的坐标。

用中心矩表示的一阶矩恒为零。

$$m_{1,0} = m_{0,1} = 0$$

对于 $p+q=2,3,4,\cdots$ 的高阶矩,可以定义规一化的中心矩

$$u_{p,q} = \frac{m'_{p,q}}{(m'_{0,0})^r},\ r = \left(\frac{p+q}{2}+1\right)$$

利用规一化的中心矩,可以获得一系列对于平移、旋转、尺度变换的矩不变量包括

$$\varphi_1 = u_{2,0} + u_{0,2}$$

$$\varphi_2 = (u_{2,0} - u_{0,2})^2 + 4u_{1,1}^2$$

$$\varphi_3 = (u_{3,0} - 3u_{1,2})^2 + (u_{0,3} - 3u_{2,1})^2$$

$$\varphi_4 = (u_{3,0} + u_{1,2})^2 + (u_{0,3} + u_{2,1})^2$$

$$\varphi_5 = (u_{3,0} - 3u_{1,2})(u_{0,3} + u_{1,2}) \times$$
$$[(u_{3,0} + u_{1,2})^2 - 3(u_{2,1} + u_{0,3})^2] +$$

$$(u_{0,3}-3u_{2,1})(u_{0,3}+u_{2,1})\times$$
$$[(u_{0,3}+u_{2,1})^2-3(u_{1,2}+u_{3,0})^2]$$
$$\varphi_6=(u_{2,0}-u_{0,2})[(u_{3,0}+u_{1,2})^2-(u_{2,1}+u_{0,3})^2]+4u_{1,1}(u_{3,0}+u_{1,2})(u_{0,3}+u_{2,1})$$

矩不变量已成功地应用于飞机形状的区分、字符识别以及景物匹配等。

参考文献

1. 余松煜,周源华,吴时光. 数字图像处理. 北京:电子工业出版社,1989

2. Jain A K. Fundementals of digital image processing. Prentice Hall Inc. ,1989

3. Gonzales R, Woods R. Digital image processing, Addison-Wesley Publishing Company Inc. , 1992

(曾建超)

tuxiang lijie xitong

图像理解系统(image understanding systems)　从输入的图像/视频中获取为完成一项或多项任务所需的视觉信息的系统,也称为计算机视觉系统。广义上,图像理解系统是要使机器具有与人或其他动物一样或类似的视觉能力,支持完成诸如在复杂环境中的导航和识别物体等工作。

图像理解通常是任务依赖的,一般而言图像理解系统由图像/视频的输入、必要的预处理过程、特征的提取与选择、分析与决策以及模型的训练与知识库等部分组成。

对图像理解而言,由于其问题本身是成像过程的逆过程,在这个过程中除了直接采用深度传感器之外,都会把深度信息丢失,同时诸如光照,材料特性、朝向、距离等信息都反映成唯一的测量值——强度,因而要从这唯一的测量值恢复上述一个或几个反映物体本质特征的参数是一个病态的(Ill-posed)过程。

对图像理解系统的探索始于 20 世纪 60 年代,早期的工作主要集中在对积木世界等简单问题(Toy Problem)的探索上,如 Roberts 等开发的三维积木理解系统。进入 70 年代之后,围绕航空、卫星图片的自动分析与理解产生了若干系统,典型的如 Brooks,Greiner 和 Binford 等开发的 ACRONYM 系统,能够完成对机场等环境的识别与理解。80 年代中后期在自动车辆驾驶的道路分析与导航系统方面产生了若干代表性的系统,典型的如卡内基梅隆大学机器人研究所完成的 Navlab 系统。近十几年来,针对人脸识别、多类物体分类、网络图像检索等任务的图像理解系统取得了很大的进展,并在搜索引擎等方面得到了广泛的应用。

目前为止,通用目的的图像理解系统还远远没有达到人们预期的目标,究其原因在于一幅图像包含的内容是异常丰富的,对于不同的目的,其理解任务也是千差万别的。与此同时在特定任务的理解方面如工业检测、文字识别、道路识别与车辆导航以及人脸检测与识别等方面已经取得了长足的进步。

D. Marr 的理论(参见**视觉计算理论**)对图像理解影响最为深远,其理论强调表示(Representation)的重要性以及从不同层次上去研究信息处理问题,在计算理论和算法实现上又特别强调计算理论的重要性。从方法上而言,近年来基于统计的方法逐渐成为图像理解系统的核心方法,这也使得系统的性能在一定程度上受到了训练数据和训练方法的影响。

目前的图像理解系统主要是从计算的角度来解决图像理解的问题,未来的趋势之一是随着脑科学、神经生理学等研究的进步,从人类和动物视觉系统借鉴更为有效的处理方式,从而实现更加通用的高效、稳定的图像理解系统。

参考文献

Marr D, Poggio T A, Ullman S. Vision: A computational investigation into the human representation and processing of visual information. MIT Press, 2010

(陈熙霖)

tuxiang linyu yunsuan

图像邻域运算(image neighborhood operation)　以图像中某一像素为中心,利用该像素及其邻域像素的灰度进行的运算。运算的结果将赋值于中心位置的像素。邻域运算的方法有空间域中的线性滤波(卷积)、非线性滤波(选择性平滑、中值滤波等)及图像相关运算等。典型的邻域运算是为了图像平滑和图像锐化(参见**图像增强**)。

模板卷积　模板卷积是一种非常有用的空域线性滤波的基本方法,可用于增强图像中的某种特征,削弱另外一些特征,检测边界,平滑噪声等。模板是一个具有特定值的阵列,当一个模板在图像中移动、相乘、求和,就完成了所谓的“卷积”操作。设 $T(i,j)$ 是一个 $m\times n$ 模板(m,n 是正整数),$f(x,y)$ 是一幅 $M\times N$ 的图像,则 T 和 f 的卷积可写为

$$g(x,y) = T * f(x,y) = \sum_{i=0}^{m-1}\sum_{j=0}^{n-1} T(i,j) f(x-i, y-j) \quad (1)$$

在一些图像处理的文献及应用中，所谓的“卷积”往往表示的是互相关计算，即

$$g(x,y) = T \cdot f(x,y) = \sum_{i=0}^{m-1}\sum_{j=0}^{n-1} T(i,j) f(x+i, y+j) \quad (2)$$

这样做只是处理上的方便，因为利用式（1）进行计算时，需要对模板进行“反折”。

非线性滤波 有几种非卷积模板的图像邻域运算是自适应的图像非线性滤波。包括中值滤波和基于图像局部统计的选择平均法和加权平均法。这类方法是根据图像窗口内的图像内容进行的邻域运算，目的在于平滑图像的同时保护边缘。

中值滤波 是一种经典的非线性低通滤波方法。以某点为中心的窗口内的像素值按从大到小排列，其中间值（当窗口内元素为偶数时可取两中间值之平均）作为该点处的输出结果。中值滤波有利于除去孤立点或线的噪声，同时保持图像的空间分辨率。但不容易去除高斯噪声。

选择平均法 只对灰度值相同或相近的像素进行平均，以免造成目标边缘的模糊。选择平均法的模板是随图像可变的模板，系数非0即1，模板大小一般取3×3，如果模板大，去噪声效果虽明显，但计算复杂，又容易抹去图像细节，造成模糊。

加权平均法 按照邻域像素灰度特殊的程度加权之后再求和，以免造成目标边缘模糊。

图像相关运算 相关操作可用于判别在一幅图像中是否存在某一已知的形状。对于一个 $M \times N$ 的图像 $I(x,y)$，可用已知形状的模板 $t(i,j)$ 通过相关运算，从图像中找出该形状，其相关定义是

$$c(x,y) = \sum_{i=0}^{m-1}\sum_{j=0}^{n-1} t(i,j) f(x+i, y+j) \quad (3)$$

其中模板的大小为 $m \times n$。使用相关运算后，在图像中和模板相匹配的位置相关函数 $C(x,y)$ 将出现峰值。即表示该位置上存在已知形状。

图像平滑 图像平滑用来减少图像中的噪声。图像平滑技术包括空域中的邻域平均法、中值滤波法和加权平均法、多图像平均法和频域中的低通滤波法。

图像锐化 图像锐化通过增强图像中的高频成分，加重图像轮廓以减少图像的模糊。在空域中，图像锐化要通过微分运算来进行。微分运算可以发现图像中灰度的变化，但不能保持图像的灰度等级。所以，基于微分的高通滤波适于边缘检测，但不一定适合一般的图像增强的要求。为了达到图像增强的目的，可以利用锐化滤波，即把微分滤波器和原图像组合起来。图像锐化也可用频域中的高通滤波来实现（参见**图像变换**）。

参考文献

1. Castleman K R. 数字图像处理. 朱志刚，等译. 北京：电子工业出版社，2002

2. Gonzalez R C, Woods R E. 数字图像处理. 2版. 阮秋琦，阮宇智，等译. 北京：电子工业出版社，2004

3. 许录平. 数字图像处理. 北京：科学出版社，2007

（朱志刚）

tuxiang moxing

图像模型（image model）

图像模型（image model） 图像的物理或数学表达及其在计算机中进行储存和加工时所采用的表示。

在**数字图像处理**中，把人眼能够看到的客观世界称为*景物*。当从某一点观察景物时，物体所发出的光线进入人的眼睛（或摄像机），在人眼的视网膜上（或摄像机的传感器上）成像，该像反映了客观景物的亮度和颜色随空间位置和方向的变化，因此它是空间坐标的函数。所谓图像，就是视觉景物的某种形式的表示和记录。我们都熟悉彩色或黑白照片。黑白（或称单色）照片记录了景物中（物体表面）每点的亮度，而彩色照片不仅记录亮度还记录颜色。随着科学技术的发展，人类不仅能够获得并记录那些人眼可见的图像信息，并且可以利用非可见光和其他手段成像，并转换成人眼可见的图像，如X射线成像、红外线成像、超声成像、核磁共振成像、激光距离成像等（参见**图像获取**）。这些成像手段使得人的视觉能力大大地增强和延伸。

图像的模型

人眼看到的景物所对应的一幅平面图像上的信息首先表现为光的强度。它是随位置坐标 (x,y)，光线的波长 λ 和时间 t 而变化的，因此图像函数可以写成

$$I = f(x, y, \lambda, t) \quad (1)$$

如果只考虑不同波长光的能量作用，那么在视觉效果上只有黑白深浅之分，而无彩色变化。这样的图像被称为*黑白图像（灰度图像）*。灰度图像模型可表示为

$$I(x,y,t) = \int f(x,y,\lambda,t)V_s(\lambda)\mathrm{d}\lambda \quad (2)$$

其中 $V_s(\lambda)$ 为相对视敏函数。

当考虑不同波长光的彩色效应时，就得到**彩色图像**。根据三基色的原理，任何一种彩色都可分解为红、绿、蓝（RGB）三种基色，所以彩色图像可表示成

$$I = \{R(x,y,t),\ G(x,y,t),\ B(x,y,t)\} \quad (3)$$

式中
$$R(x,y,t) = \int f(x,y,\lambda,t)R_s(\lambda)\mathrm{d}\lambda$$
$$G(x,y,t) = \int f(x,y,\lambda,t)G_s(\lambda)\mathrm{d}\lambda$$
$$B(x,y,t) = \int f(x,y,\lambda,t)B_s(\lambda)\mathrm{d}\lambda$$

其中 $R_s(\lambda)$，$G_s(\lambda)$ 和 $B_s(\lambda)$ 分别是 R、G、B 三种基色的视敏函数。

当图像内容随时间变化时，它们被称为**时变图像**或**运动图像**。运动物体的图像、电影、电视等都是运动图像。当图像内容不随时间变化时，称为**静止图像**。对黑白静止图像而言，其函数为

$$I = f(x,y) \quad (4)$$

由于人眼和其他成像系统视野有限，因此图像在空间上是有界的。其取值区域通常定义为矩形。图像函数在某一点的值通常称为亮度，一般用正实数表示。亮度或灰度的数值是有限的，即 $0 \leqslant f(x,y) \leqslant B_m$ 其中 B_m 为最大亮度。

采样和量化

只有将上面定义的图像转换成**数字图像**，才能用数字计算机进行处理。采样是图像 $f(x,y)$ 进入计算机的第一步。采样过程包括将一个 $(M \times N)$ 的采样网格覆盖在图像 $f(x,y)$ 上，然后度量每个方格的平均亮度，这样，图像 $f(x,y)$ 就可以用计算机中的一个 $M \times N$ 矩阵 $[f(m,n)]$ 来表示。$f(m,n)$ 为矩阵中的一个元素。

采样中的主要问题是采样密度应该多大，才不致丢失原图的信息。采样定理回答了这个问题。对于在 x 和 y 方向上截止频率分别为 ξ_x 和 ξ_y 的有限带宽图像，在 x 和 y 方向上的采样间隔应满足

$$\Delta x \leqslant 1/(2\xi_x),\ \Delta y \leqslant 1/(2\xi_y) \quad (5)$$

如果 $f(x,y)$ 在 $x \in [0,X]$，$y \in [0,Y]$ 内有定义，并以 Δx 和 Δy 为间隔取样，则沿 x 和 y 方向的取样点数为：$M = X/\Delta x, N = Y/\Delta y$。

对图像 $f(x,y)$ 的采样完成后，得到图像各点的采样值 $f(m,n)$。但是，在输入数字计算机之前，对 $f(m,n)$ 还要进行量化。具体地说，就是要在样本值的取值范围内进行分层（分成 k 个区间），然后用单个值（逻辑值）来代表各层内所有可能的亮度（物理值）。根据计算机内整数存放的惯例，一般可以把样本的取值分成 $k = 2^l$ 个层次，例如 $l = 6$,7 或 8，即把矩阵中每个元素的灰度值分成 64、128 或 256 个层次。一般来讲，层次越多，由量化了的取样值恢复的实际图像（如显示的图像）越接近原图，看上去越满意。当层次不够时，在图像中缓慢变化的区域将出现原图像所没有的假轮廓。

最简单的量化方案是均匀量化，即把样本值的整个取值范围均分成 k 个子区间。如果样本值在某个取值范围内频繁出现，而在其他范围内很少出现，就可以在样本值频繁出现的范围内采用小量化区间的密集量化，而在其他地方量化得粗糙一些，这样可以在不增加量化层次的条件下，减少实际引入的量化噪声，这就是非均匀量化。

亮度和对比度 在均匀量化中，假如量化级为 256，则 0 代表最小亮度（黑），255 代表最大亮度（白）。亮度值在 0～255 范围内呈线性关系。例如，灰度值为 42 的点是灰度值为 21 的点的亮度的两倍。图像对比度反映了图像中亮度的动态范围。如在 256 级灰度下，一幅具有灰度范围为[50,100]的图像就比一幅灰度范围为[30,210]的图像的对比度低。

像素和分辨率 尺寸为 $M \times N$ 的数字图像矩阵 $[f(m,n)]$ 中的每个元素称为图像元素，简称**像素**，M 表示水平行数，N 表示垂直列数。像素是数字图像可以赋值的最小单位。每个像素对应的图像物理尺寸由采样决定。术语“**分辨率**”描述了数字图像所能表示的空间细节的程度。图像分辨率一般指数字图像中像素的个数 $(M \times N)$。对分辨率的要求常常是和应用有关的。例如，为了产生满意的电视效果，分辨率（$M \times N$ 的大小）应不少于 500×500。一般计算机显示的分辨率为 640×480。当然，有的应用需要较高的分辨率为 1024×1024，而另一些只要较低的分辨率为 32×32 就够了。图像获取和图像显现中有关分辨率的定义和具体设备相关。

数字图像的表示

针对图像的不同含义或不同的处理要求，数字图像常用的表示形式有：

灰度图像（或称单色图像） 每个像素用一个字节表示，256 级灰度可表示高质量的图像。而且表示方法也比较紧凑（八位二进制数即一个字节）。

512×512×8 位的图像和标准电视分辨率相近,但有更好的灰度层次。有的数字图像特别是医学图像,需要用十六位二进制数表示,而且常常是有正负号的。这种有符号的表示对一些图像处理的中间结果也非常有用,它可以避免数据的上溢和下溢。

浮点数图像 实数可以用作图像处理的中间结果,它既可以保留运算中产生的小数部分,又可以表示很大的数据范围。复数也可用于图像的表示,主要用于数字图像转换到其他域中的表示,如傅里叶变换域。

彩色图像 彩色图像的采样和单色图像相似,但需要分别测量和采样红、绿、蓝三基色分量,因此彩色图像的数据量是单色图像的三倍。在实际应用中,彩色图像几乎总是数字化成各为八位的三个分量。这种形式亦称为 24 位彩色图像,意即每个像素需用 24 位表示;有时也说这种图像可以表示出 $2^{24}\approx 16$ M(兆)种颜色。

二值图像 在每个像素上只表示某种特性的有或无,即逻辑表示上的真或假,称作二值图像。通常二值图像的每个像素只用 1 个二进制位(bit)表示,一个字节表示 8 个像素(一般按行储存)。在实用中,图像的一行像素用整数个字节表示,典型的应用有文本图像处理、图像打印以及图像表示等。

参考文献

1. Castleman K R. 数字图像处理. 朱志刚,等译. 北京:电子工业出版社,2002

2. Gonzalez R C, Woods R E. 数字图像处理. 2 版. 阮秋琦,阮宇智,等译. 北京:电子工业出版社, 2004

3. 许录平. 数字图像处理. 北京:科学出版社, 2007

(朱志刚)

tuxiang peizhun

图像配准(image registration) 在几何上对齐两个或两个以上在不同的时间、或从不同的视角、或用不同的传感器所获取的同样场景的图像的过程。实行几何对齐的两个图像分别称为参考图像和目标图像。由于上述不同成像条件,图像间引入了外观差异。图像配准是许多图像分析任务的关键步骤,其中包括图像融合、变化检测以及多通道图像恢复。在这些任务中,最后所需的信息来自各种数据源的组合。

图像配准广泛应用于遥感,医学成像,制图和**计算机视觉**等领域。在遥感应用中,图像配准是多光谱分类,环境监测,变化检测,图像拼接,气象预报,超高分辨率图像生成,地理信息系统(GIS)整合等的关键。在医疗成像中,图像配准通常用在计算机断层成像技术(CT)和核磁共振(MRI)数据的融合,以获得更全面的患者信息;将患者的数据和解剖学图谱进行比较,以便更好地监测肿瘤生长,核查处理的效果。在制图学中,图像配准可用在地图的更新。在计算机视觉中,图像配准是立体匹配,目标定位,自动质量控制,运动分析等非常重要的步骤。

图像配准的主要类别

一般来说,根据图像采集方式(不同的视角,不同的时间,不同的传感器或不同的来源),图像配准可分为四大类:多视角配准,多时间配准,多模态配准和像模配准。

多视角配准 指的是从不同视角获取的同一场景的两幅或多幅图像的几何对齐。其目的是为了获取更大视角的二维视图甚至三维场景表示。应用的例子包括:遥感成像中的图像拼接,计算机视觉中的形状恢复和集成。

多时间配准 指的是在不同时间获取的同一场景的两幅或多幅图像的几何对准。在遥感图像和医学图像处理中,多时间配准经常是定期的操作。其目的是检测和评估在不同时间,并很可能是在不同条件下获取的图像之间的变化。应用的例子包括:遥感监测中的全球土地使用、景观规划,计算机视觉中为实现安全监控或运动跟踪的自动变化检测,医疗成像中治疗愈合或肿瘤演变的监测等。

多模态配准 指的是使用不同传感器获取的同一场景图像的对准。其目的是融合来自不同来源的信息,以获得更丰富或更详细的景物表示。应用的例子包括:遥感中不同特性的传感器信息的融合,以便获得更好的空间分辨率;彩色/多光谱图像的融合以便提高光谱分辨率或减少雷达图像的云层和太阳光照的影响;医疗成像中融合人体解剖结构的传感器记录,如核磁共振(MRI)、超声或 CT,与监测功能和代谢的身体活动的传感器,如正电子发射断层(PET)、单光子发射计算机断层显像(SPECT)或磁共振波谱(MRS)。

像模配准 指的是场景图像和场景模型的配准。这里模型可以是同一场景的计算机表示,例如在地理信息系统(GIS)中的地图或数字高程模型(DEM);也可以是另一个具有类似内容的场景(或另一位病人),或者是一个“平均”的标本等。其目的是将所获得的图像锁定在场景或其模型上,以便

对它们进行比较分析。应用实例包括：遥感中将航空或卫星图像和地图或其他 GIS 分层的对准；计算机视觉中的目标模板与输入图像的匹配、自动质量检验；医学影像中患者图像和数字解剖图谱的对比、标本分类。

图像配准的主要步骤

由于需要配准的图像的多样性以及各种各样的退化原因，不可能设计一个通用方法适用于所有的配准任务。每种方法不仅应考虑到图像之间的几何变形，而且也要考虑到它们之间的辐射变形和噪声腐蚀以及配准的准确性要求和应用相关的数据特征。虽然如此，大多数的配准方法基本包括以下 4 个步骤：

（1）特征检测：可以用手工或最好是自动的方法检测突出或独有的特征，如封闭边界区域、边缘、轮廓、线交叉点、弯道等。为了作进一步处理，这些特征可以用代表它们的控制点（control points）来表示，如重力中心、线段的结束、独特的点等。

（2）特征匹配：这一步将在目标图像和参考图像中检测出的特征点之间建立对应的关系。为了达到可靠的匹配，可以采用不同的特性描述、相似性度量以及相应的空间特征表示。

（3）变换模型估计：在这一步需要估计出用于对准目标图像和参考图像的映射函数的类型和参数。一般来说，映射函数的参数是根据对应点的匹配计算出来的。

（4）图像重采样和变换：目标图像通过映射函数得到几何变换（参见**图像几何运算**）。在非整数坐标上的图像数值可以通过适当的插值技术获得。

每个配准步骤的实施，都有其典型的问题。首先需要确定的是，对于给定的任务，什么样的特征是适当的。这些特征通常应当对应鲜明的物体标志，而且容易被检测。一般情况下，特征需具备物理意义的可解释性，如物体的边界点、特别图案的中心等。在参考和目标图像中检测出的特征应当有足够的共同元素以供匹配，即便是两幅图像不包括完全同样的场景，或有不同的物体遮挡，或其他意想不到的变化。检测方法也应具有良好的定位精度，不应过于受图像质量下降的影响。

在特征匹配步骤，可能产生的问题往往源于不正确的图像特征检测或可能发生的图像退化，其中包括图像的几何变化和外观变化。由于不同的成像条件或不同传感器不同的光谱灵敏度，物理上对应的图像特征可能很不相似。因此在设计特征的描述和相似性的度量时，需要考虑这些因素。特征的描述最好具有独立于图像退化的不变性。同时，它们还必须具有足以区分不同特征的能力以及足够的稳定性，以免被轻微的变化和噪声影响。基于不变量的空间匹配算法应是稳健和高效的。

映射函数的类型选择应当根据有关成像过程的先验知识和可能的图像退化（变形）来确定。如果没有先验的信息可用，选择的模型应当足够灵活和通用，以应付可能的图像变形。在图像变换模型估计中，应当考虑的因素包括特征检测方法的准确性、特征对应的可靠性以及可以接受的变换误差等。此外，特别需要考虑图像之间的哪些不同必须通过或不能通过图像配准来消除。例如，如果我们的目标是图像变化的检测，我们所要寻求的变化就不能被图像配准消除。这个问题是非常重要也是非常困难的。

最后，重采样技术的适当类型的选择取决于插补精度要求和计算复杂性的平衡。最近邻或双线性内插在大多数情况下是足够的，但有些应用需要更精确的方法。

参考文献

1. Brown L G. A survey of image registration techniques. ACM Computing Surveys, 1992, 24: 326-376

2. Zitová B, Flusser J. Image registration methods: a survey. Image and Vision Computing, 2003, 21: 977-1000

3. 李玲玲，黄其民，李保. 多传感器图像配准方法综述. 光盘技术，2007，(2)

4. 刘松涛，杨绍清. 图像配准技术研究进展. 电光与控制，2007，14(6)

（朱志刚）

tuxiang quyu biaoshi

图像区域表示（image region representation） 把图像中的某些区域用合适的数据结构或数学式表示的方法。图像的区域表示方法有 *y* 轴表示及四叉树表示等。

***y* 轴表示** 这是一种将区域置于 *x-y* 平面内并用相应的坐标排列来表示区域的方法。*y* 轴表示是一个含有多个子表的表格。主表的数字对应于区域元素所在的 *y* 坐标，子表的第一个数字是同一 *y* 坐标下第一个区域元素的 *x* 坐标，子表的第二个数字是该区域最后那个元素的 *x* 坐标。若该区域的右面还有别的区域，子表依上述规则继续排列。图 1 是一

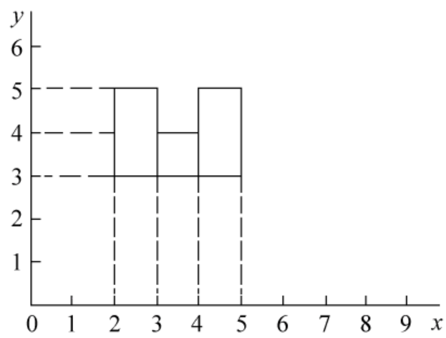


图 1　y 轴表示

个用 y 轴表示的例子。图中的区域可表示为[(3,4,5)(2,5)(3,4)(2,3,4,5)]。

四叉树表示　这是对图像所在区域用树形结构表示的一种方法。这种方法在数字图像处理、景物理解以及计算机图形学等有广泛的用途。

四叉树是对图像所在平面进行递归剖分形成的。首先以包围整幅图像的一个矩形为根结点,然后将该矩形作 4 等分,形成 4 个子结点,并加以编号。如果某一结点对应的矩形全部被图像所充满,则称为黑结点;如果某一结点对应的矩形无图像存在,称为白结点;如果某一结点对应的矩形部分地被图像占有,则称为灰结点。黑白两种结点不再细分;而灰结点则需作进一步剖分。这一过程递归进行,直至全部为黑白两种结点或者达到预先规定的精度为止。

应该指出,四叉树是图像所在区域的表示。一幅图像及其四叉树表示见图 2。

参考文献

1. 赵荣椿,等. 数字图像处理导论. 2 版. 西安:西北工业大学出版社,1995

2. Sonka M, Hlavac V, Boyle R. Image processing, analysis and machine vision. Chapman & Hall, 1993

(曾建超)

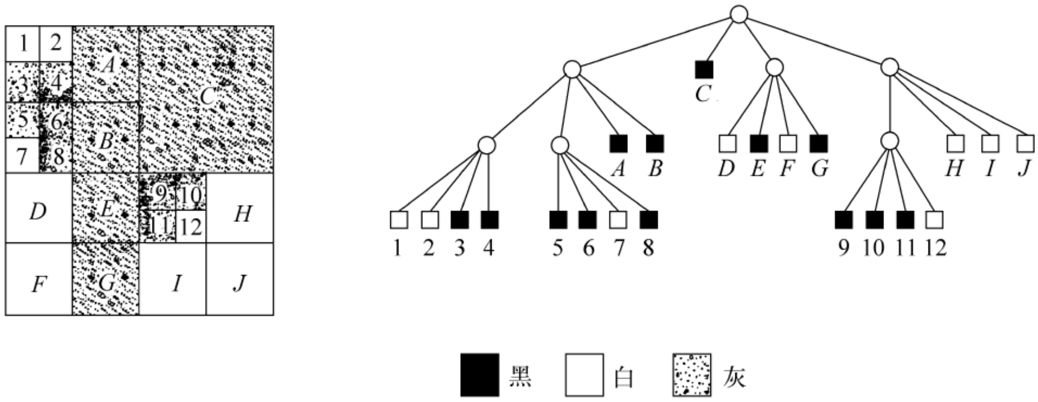


图 2　四叉树表示

tuxiang shuiyin

图像水印(image watermarking)　是将特制的标记隐藏在数字图像中的一种技术(标记一般不可见,但有些情况下是可见的,例如 logo。可见与不可见的要求取决于应用目的)。图像水印主要用于确定版权所有者、识别购买者、跟踪侵权行为、认证数字内容来源的真实性、提供关于数字内容的其他附加信息等。

图像水印的信息处理过程可分为嵌入过程和检测过程。嵌入过程接收两个输入,即要编码为水印的信息和要嵌入水印的载体数据。嵌入过程的输出通常会被传输或记录。检测过程将设法判断给定数据(可能是有水印的载体数据或未经水印嵌入处理的载体数据)中是否存在水印,若存在,则输出水印编码的信息。可以通过嵌入有效性、保真度、数据有效载荷、检测要求(盲检测/含辅助信息检测)、虚警率、鲁棒性(脆弱水印则不需要鲁棒性)、安全性、计算成本等特性来评价水印系统的性能。同时,也可以根据具体的应用需求调整水印系统的各个特性使它们达到一定的平衡(某个特性的改善通常是通过牺牲其他特性的性能而得到的)。

在图像中嵌入水印的可能性来自于图像数据的视觉冗余性。从视觉信号处理的角度,水印信息可以视为在强背景(原始图像)下叠加的一个弱信号。由于人的视觉系统(HVS)分辨率受到一定限制,只要叠加的弱信号的幅度低于 HVS 的对比度门限,HVS 就无法感觉到信号的存在。因此,通过对原始图像做有限制的改变,就有可能在不改变视觉效果的情况下嵌入一些信息。

图像水印的算法可以分为两大类,即空间域水

印算法和变换域水印算法。空间域水印算法直接在原始图像上隐藏信息,如最低有效位算法(通过图像数据的最低几位来隐藏信息)、Patchwork 算法(随机选择图像点对,通过增加一点的亮度同时降低另一点的亮度值的方式隐藏信息)、基于空域分块的方法(通过改变图像块的均值来隐藏信息)等。变换域水印算法是先对原始图像进行某种正交变换(如**离散余弦变换**、离散**小波变换**、离散**傅里叶变换**等),然后通过修改某些变换参数来隐藏信息。与空间域水印算法相比,变换域水印算法的优点有:①在变换域中嵌入的水印信号能量可以散布到空间域的所有位置上,有利于保证水印的隐藏性;②在变换域,人类视觉系统的某些特性可以更方便地结合到水印编码过程中;③变换域方法可与图像压缩标准相兼容,从而实现在压缩域内的水印算法,同时也能抵抗相应的有损压缩。

参考文献

1. 杨义先,钮心忻. 数字水印理论与技术. 北京:高等教育出版社,2006

2. 孙圣和,陆哲明,牛夏牧,等. 数字水印技术及应用. 北京:科学出版社,2004

3. Cox I J, Miller M L, Bloom J A. 数字水印. 王颖,黄志蓓,等译. 北京:电子工业出版社,2003

(邱慧军)

tuxiang tuihua

图像退化(image degradation) 图像在形成、传输和记录过程中,由于成像系统、传输介质和设备的不完善,导致图像质量下降,使图像变得模糊的现象。引起退化的原因很多,如光学系统的像差、衍射、非线性、几何畸变、成像系统与被摄体的相对运动、大气的湍流等。此外退化图像还可能叠加有各种随机噪声,如相纸上的斑点等。图 1 将退化综合为一模型,与其对应的数学表达式为

$$f(x,y) = H[g(x,y)] + n(x,y) \qquad (1)$$

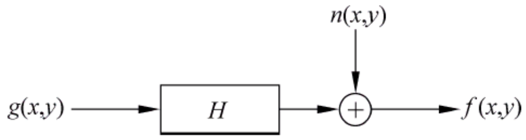


图 1 图像退化过程的模型

图 1 的模型表示,一幅模糊图像 $f(x,y)$ 可以看作是原始图像 $g(x,y)$ 经过算子 H(把图像的退化过程模型化为一个施加于 $g(x,y)$ 的一个算子或运算)作用后再叠加噪声 $n(x,y)$ 的结果。

参考文献

1. 贾永红. 数字图像处理. 2 版. 武汉:武汉大学出版社,2010

2. 赵荣椿,等. 数字图像处理导论. 2 版. 西安:西北工业大学出版社,1995

(李树青)

tuxiang Weina lübo fuyuan

图像维纳滤波复原(image restoration by Wiener filtering) 满足最小均方误差准则的一种图像复原方法。由于大多数退化图像都含有噪声,使用反向滤波法(参见**图像反向滤波复原**)经常导致复原失败。针对这种情况,可以应用维纳滤波。假定退化模型为线性并且空间位移不变。原始图像 $g(x,y)$、退化图像 $f(x,y)$ 均为平稳随机函数,$n(x,y)$ 为幅值有限的加法噪声,点扩展函数 $h(x,y)$ 是已经确知的函数,那么可求得复原图像的估值 $\hat{g}(x,y)$(参见**图像反向滤波复原**),使其与原始图像 $g(x,y)$ 之间的均方误差 e^2 最小,即

$$e^2 = E[\,|g(x,y) - \hat{g}(x,y)|^2]$$

这种满足均方误差最小的复原方法就是维纳滤波。

维纳滤波的复原公式如下(这是最常用的维纳滤波的频域解)

$$\hat{G}(u,v) = \frac{1}{H(u,v)} \times \frac{[H(u,v)]^2}{[H(u,v)]^2 + r\,S_n(u,v)/S_g(u,v)} F(u,v)$$

$$u,v = 0,1,\cdots,M-1 \qquad (1)$$

式中,$H(u,v)$——点扩展函数的傅里叶变换;

$F(u,v)$——退化图像的傅里叶变换;

$\hat{G}(u,v)$——复原图像估值的傅里叶变换;

$S_g(u,v)$——图像功率谱(图像自相关矩阵的傅里叶变换);

$S_n(u,v)$——噪声功率谱(噪声自相关矩阵的傅里叶变换)。

这里 $S_g(u,v)$ 和 $S_n(u,v)$ 是从属于某一类的许多幅图像统计而得的参数,而 r 是供选择的参数。当 $r = 0$ 时,式(1)变为

$$\hat{G}(u,v) = F(u,v)/H(u,v) \qquad (2)$$

这就是反向滤波。

当 $r = 1$ 时

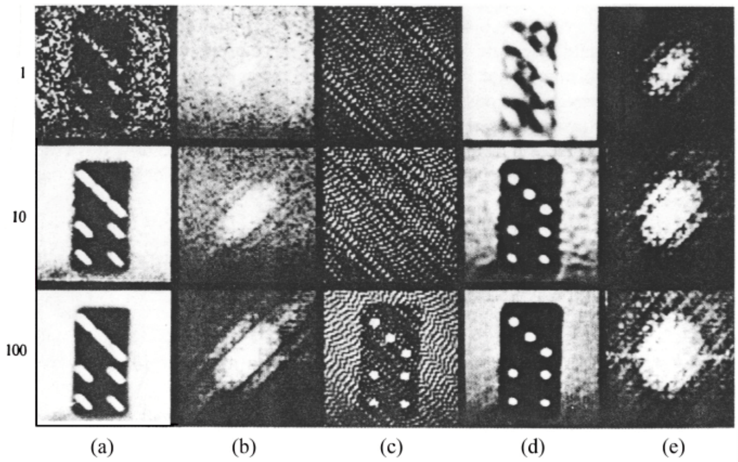
$$\hat{G}(u,v) = \frac{1}{H(u,v)} \times \frac{[H(u,v)]^2}{[H(u,v)]^2 + S_n(u,v)/S_g(u,v)} F(u,v)$$

$$u,v = 0,1,\cdots,M-1 \qquad (3)$$

这种情况称之为标准维纳滤波。

r 也可以取其他值。通过调整 r 可获得不同的原始图像的近似值 $\hat{G}(u,v)$，以满足均方误差减少的条件。这称之为**带参数的维纳滤波**。

采用维纳滤波时，信号与噪声的比对复原图像影响比较小。图 1 给出了用维纳滤波和反向滤波对模糊图像进行复原处理的例子。可以看到，当噪声比较大时，维纳滤波复原的图像质量优于反向滤波复原的图像质量。



（a）退化图像（由上到下信噪比为 1、10、100）；
（b）退化图像的傅里叶频谱；
（c）反向滤波复原的图像；
（d）维纳滤波复原的图像；
（e）维纳滤波复原后图像（d）的傅里叶频谱

图 1　维纳滤波和反向滤波对模糊图像进行图像复原的比较

（此图取自参考文献 3）

参考文献

1. 贾永红. 数字图像处理. 3 版. 武汉：武汉大学出版社，2015.

2. 赵荣椿，等. 数字图像处理导论. 2 版. 西安：西北工业大学出版社，1995

3. Rafael C，Gonzalez/Paul Wintz. digital image processing. 2nd ed. Addison-wesley，1987

4. Jain A K. Fundementals of digital image processing. Prentice-Hall Inc.，1989

（李树青）

tuxiang wenli chuli

图像纹理处理（image texture processing）

对图像纹理进行描述、分析、生成、应用等处理过程的理论、方法和技术。纹理是人眼对自然界中物体表面的自相似和重复结构，或相似物体所组成的表面集合的一种视觉认知现象。它是人们区分自然界中不同物体、物体集合的视觉特征之一。被广泛应用于**数字图像处理**中物体的分割和识别，以及**计算机图形学**中的**真实感图形生成**。视觉中的纹理现象十分复杂，目前还缺乏公认的数学描述方法和严格的定义。但一般认为，大部分纹理存在一些共性。首先，自然纹理具有自相似的属性，如远处的树林，地上的落叶，河滩上的鹅卵石等，这一属性使得很多纹理可以用多尺度的小波或统计方法来描述；其次，许多人工产生的纹理具有多种重复结构，如条纹布，多窗建筑的外立面等。这一属性使得很多纹理可以用频谱或句法来描述；再次，一些复杂的纹理，如孔雀羽纹等许多自然界中多种颜色所构成的纹理，有些符合一些统计规律，可以用重复单元的句法来描述，有些则难以描述。纹理的表示和模型是图像纹理研究的基础，纹理模型既可以用来分割和识别纹理，也可以根据纹理模型来生成纹理。

总体来说图像纹理的应用可分为两大部分：①纹理的分析，其中包括纹理分割和纹理识别；②纹理生成，其中包括纹理合成（参见**纹理合成**）和纹理映射（参见**纹理映射**）。

纹理分割是指基于纹理的图像分割，即分割图像中不同纹理的区域。首先采用纹理分析的方法，得到图像中纹理的描述，再根据该描述区分图像中的不同纹理区域。因此，纹理分割一般同时得到图

像中纹理区域的描述。基于该描述可以进一步识别纹理和理解图像内容。例如,对于一幅具有天空、草地、树林的图像,首先可以采用 Markov 随机场的纹理分析方法,对三种纹理进行分析,获得这些纹理的模型和参数;其次,根据分析的结果,将图像分割为天空、草地、树林等区域;最后,在纹理描述和分割的基础上,对图像进行理解。如果一幅图像包括天空、草地、树林,则该图像可能是一个公园,也可能是草原中的一角。但是,如果图像中还有建筑的纹理,则该图像是城市中的一个公园的概率较大。在医学图像中,纹理分割方法可以区分正常和病变的组织,用于自动或辅助诊断疾病。**纹理识别**是指基于纹理特征抽取和表示,对图像中的纹理区域进行描述,并和已知的纹理进行匹配,从而对区域纹理进行分类和识别,它是基于纹理图像理解的基础。纹理描述主要是依据纹理的自相似和重复结构两大共性,来描述图像中的灰度分布和变化规律。彩色图像的纹理描述一般将其转换为灰度图像后进行,但部分基于统计和纹理结构分析的方法,也可以描述彩色图像的纹理。

纹理描述的主要方法有统计分析法、信号分析法和结构分析法三种。其中统计分析法计算纹理的一阶或二阶统计量,如灰度直方图、灰度梯度直方图、灰度共生矩阵等;或者假设纹理图像中任一像素的灰度值和其邻域像素的灰度值存在某种概率关系,采用 Markov 随机场、Gibbs 随机场等方法,对纹理进行描述。信号分析法是指采用信号处理的方法,对纹理进行特征表示,比较传统的有傅里叶变换,小波变换等,其中 Gabor 小波由于简单且能够得到纹理的朝向等特征,在纹理描述中有着广泛的应用。结构分析法主要用于人造物体的纹理描述。和自然纹理不同,人造纹理一般具有重复单元。例如草地、卵石滩等自然纹理具有自相似的结构,但是并不存在重复单元,而砖墙等人造物体的纹理却具有重复单元。结构分析法通过分析这些重复单元的特征、大小、朝向等重复单元的排列方式,而获取纹理的描述。

由于图像本身的复杂性,图像中物体的纹理包含颜色、灰度、视角等各种变化,一般难以得到完全精确的纹理描述,因此,在实际应用中,需要综合运用**机器学习**和**模式识别**等多种方法,根据已知纹理的学习和当前纹理描述的结果,对纹理进行分类,来获得当前图像纹理区域的类别。

参考文献

1. Ojala T, Pietikainen M, Maenpaa. Multiresolution gray-Scale and rotation invariant texture classification with local binary patterns. IEEE Trans Pattern Analysis and Machine Intelligence, 2002, 24(7): 971-987

2. 马莉,范影乐. 纹理图像分析. 北京: 科学出版社,2009

(陶霖密　徐光祐)

tuxiang xingtaixue yunsuan

图像形态学运算(image morphological operation)　基于数学形态学的图像运算。它通过图像中被处理对象和结构元素的相互作用来获取该对象(物体)的拓扑或结构信息。数学形态学是研究形状和结构的学科。在图像处理中,形态学运算采用不同于空域滤波和频域滤波的形态滤波方法,也就是通过图像中被处理对象和结构元素的相互作用以得到更本质的形态。

在图像处理时,利用形态学基本概念可以改善图像质量。例如,当希望获得人脸的线画图时,由于光照等原因,人脸的边缘会有噪声。使用传统图像分割技术,人脸轮廓可能不光滑。利用形态学方法虽然不能产生新的信息,但可根据人脸轮廓一般是光滑曲线这个先验知识,通过平滑除掉边界的凸起和凹陷。

形态学的概念也可以用来描述和定义图像的各种几何参数和特征,如区域数目、面积、周长、连通度、颗粒度、骨架和方向性等。

形态学运算常用于二值图像。实际上,早期的数学形态学只关心位置而不考虑灰度。现在形态学运算已扩展到灰度图像。

腐蚀和膨胀　大部分形态学运算都定义在腐蚀和膨胀两个基本运算的基础上。在二值图像的形态滤波中,它们分别为“结构差”与“结构和”运算。假定处理对象 X 和结构元素 B 为二维欧氏空间中的集合。它们的基本关系为:

B 包含于 X,记为 $B \subset X$,见图 1(a);

B 击中 X,即 $B \cap X \neq \varnothing$,记为 $B \uparrow X$,见图 1(b);

B 不击中 X,$B \cap X = \varnothing$,记为 $B \subset X^c$,X^c 为 X 的补集,见图 1(c)。

把结构元素 B 平移 $\boldsymbol{x}$($\boldsymbol{x}$ 是一向量)以后得到的元素记为 $B_{\boldsymbol{x}}$。例如在图 2 中,当 B 为 $\{(0,0),(1,0),(0,1)\}$ 而 $\boldsymbol{x}=(2,2)$ 时,$B_{\boldsymbol{x}}$ 为 $\{(2,2),(3,2),(2,3)\}$。

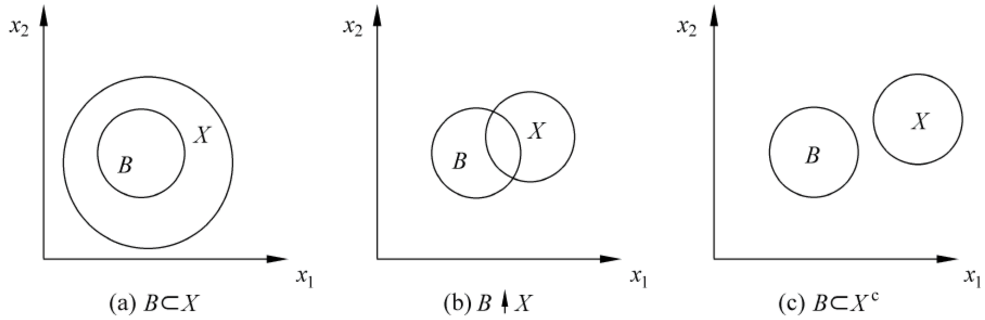


图 1　形态学运算的基本关系

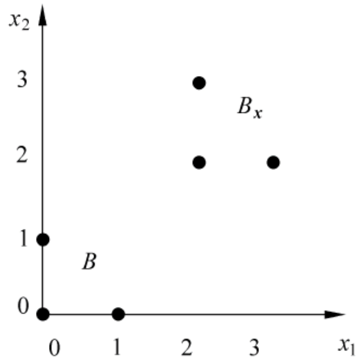


图 2　结构元素的平移

当结构元素被平移 $\boldsymbol{x}$ 后，使 B_x 包含于 X 的所有的平移量 $\boldsymbol{x}$ 的集合称为 X 被 B 腐蚀的结果，用公式表示为

$$E(X) = \{\boldsymbol{x};\ B_x \cap X\} \text{ 或记为 } X\Theta B_x$$

使 B_x 击中 X 的所有的平移量 $\boldsymbol{x}$ 构成的集合称为 X 被 B 膨胀的结果。用公式表示为

$$D(X) = \{\boldsymbol{x};\ B_x \cap X \neq \varnothing\} \text{ 或记为 } X \oplus B_x$$

腐蚀和膨胀互为对偶，用公式表示为

$$(X\Theta B)^c = X^c \oplus B$$

这里，上标“c”表示取补集，即 B 对 X 的腐蚀的补集等于 B 对 X 补集的膨胀。例如，河岸的补集为河面，那么对河岸的腐蚀等效于河面的膨胀。在有些情况下这种对偶关系非常有用。例如，某个图像处理系统用硬件实现了腐蚀，那么膨胀可以通过先将求反的原图进行腐蚀运算再将结果求反获得，而不必要再构成膨胀的硬件。

开和闭　在腐蚀和膨胀的基础上，常用的形态运算（变换）有结构开和结构闭、击中击不中变换、细化和粗化、边界和骨架等。这里仅就结构开和结构闭运算做如下解释。

用结构元素 B 对处理对象 X 先腐蚀后膨胀的结果称为 B 对 X 的开，即

$$\mathrm{OPEN}(X) = (X\Theta B) \oplus \check{B}\text{，记为 } X_B\text{。}$$

这里，$\check{B}$ 是 B 的对称集，即 $\check{B}$ 是把 B 中的元素改变符号（由正变负，由负变正）的集合。例如，$B = \{(0,0),(1,0),(0,1)\}$，则 $\check{B} = \{(0,0),(-1,0),(0,-1)\}$。

而用结构元素 B 对处理对象 X 先膨胀后腐蚀的结果称为 B 对 X 的闭，即

$$\mathrm{CLOSE}(X) = (X \oplus \check{B})\Theta B\text{，记为 } X^B$$

一般来说，结构开能去除孤立的小点、毛刺和小桥，而结构闭则能填平小湖、弥合小裂缝。

灰度图像的形态滤波　形态运算可用于一般的灰度图像。一种简单方法是将灰度图像二值化，即将所关心的灰度变为 1，其他变为 0。更好的方案是定义和二值形态运算略有不同的“灰值形态运算”。灰值腐蚀和膨胀分别用于灰度图像区域边界的突起抹平和下凹填补。同二值图像一样，同样可以在腐蚀和膨胀运算的基础上定义其他运算。

参考文献

1. Castleman K R. 数字图像处理. 朱志刚，等译. 北京：电子工业出版社，2002

2. Gonzalez R C, Woods R E. 数字图像处理. 2 版. 阮秋琦，阮宇智，等译. 北京：电子工业出版社，2004

3. 许录平. 数字图像处理. 北京：科学出版社，2007

（朱志刚）

tuxiang xiufu

图像修复（image inpainting）　根据图像现有的信息自动恢复丢失的信息；或者去除图像的部分内容同时自动填补该区域，使观察者无法觉察图像曾经有丢失或缺损，或已被修复的痕迹的图像复原

技术。图像修复就是将待修复区域周围的信息填充到待修复区域中。目前有两大类图像修复技术：一类是修复小尺寸缺损的数字图像修补(inpainting)技术；另一类是用于填充图像大块丢失的图像补全(completion)技术。修复小尺寸缺损图像首先观察分析图像现有部分的内容并建立图像的数据模型，其后要建立起数学模型，将修补问题转换为一个泛函数求极值的变分问题。解决中大块丢失信息的图像补全技术是将图像分解为结构部分和纹理部分，其中纹理部分用纹理合成方法填充。纹理合成补全技术不但可以填充任意大小的丢失块，还可以修补部分细节。它的基本思想是首先在图像丢失块的边界上任选一个像素点名，以该点为中心设定大小为 3×3 或 9×9 的模板，然后在已知图像区域内找出与模板的最优匹配块，用最优匹配块填充即可。

在许多实际应用中，往往为了某种特殊的目的，而移走数字图像上的目标物体或文字；又由于不希望观察者觉察出图像中有物体或文字被移走，为了保证图像信息的完整性，需要对这些受损图像进行填充修补。基于纹理合成的图像补全技术能达到这一目的。

图 1 示出数字图像修补技术应用于去除老照片中有划痕的修复效果。其中(a)为原始照片，(b)为被损照片(其中白色划痕三角形是待修补的区域)，(c)是修复后的结果。图 2 示出小尺度缺损图像的修复图片。图 3 示出移走数字图像上的人的影像后补全全景的照片。

参考文献

张红英，彭启琮. 数字图像修复技术综述. 中国图像图形学报，2007，12(1)

(李树青)



(a) 原始照片　(b) 被损待修复照片　(c) 修补复原的结果

图 1　去除老照片中划痕的效果图



(a) 原始照片　(b) 受损照片　(c) 修补复原后的结果

图 2　修复照片中丢失块的效果图



(a) 原始图像　(b) 想要移走的人影像　(c) 补全全景图像

图 3　移走数字图像上的人的影像后修复的图像效果图

tuxiang xulie chuli

图像序列处理(image sequence processing)

以图像序列为输入,检测、估计和跟踪关于物体的结构信息与运动的过程和方法。其目标是为识别物体运动提供在“时间-空间”域中的特征描述。物体的三维空间运动造成二维图像中亮度分布的变化,使得视觉系统有可能在物体与背景颜色相似的情况下,检测和跟踪物体。例如,青蛙利用运动信息检测有保护色的昆虫。图像序列处理研究通过图像亮度信息的变化来检测和估计物体运动的几何信息的方法和模型。由于在三维空间的物体投影到二维图像时,深度信息的丢失以及成像条件的变化对亮度分布的干扰,造成了处理中可能存在病态问题。

图像序列处理在各领域有广泛应用。可用于处理以下问题:

(1) 利用运动来检测和分割物体:如视觉监控中,从摄像机拍摄的图像序列中检测运动物体以便进一步分析其行为;医疗诊断中,从人眼视网膜的间隔几个月的不同成像中检测出视网膜组织的萎缩等。

(2) 得到物体的运动描述或几何结构:如在军事中,通过对目标的持续跟踪来辅助瞄准;在运动捕获应用中,通过获取的运动信息来驱动卡通人物的动画;在人机交互中,利用得到的运动描述来识别用户动作;在摄影测量中,通过航拍的图像序列确定被摄物体的大小、形状和空间位置等。

(3) 减少冗余度:图像序列中存在大量的时空冗余信息,通过对图像序列的压缩编码,以降低时空冗余度,便于存储和传输。

图像序列处理研究的主要内容有:

(1) 运动检测和分割　分离场景中的运动物体与静止物体(参见**运动检测**)或不同的运动物体。

(2) 运动估计　估计图像序列中相邻帧间的运动信息(参见**运动估计**)。运动估计关注的目标可以是整幅图像(即估计全局运动),每个像素(即估计运动场)或图像中的特定部分(即某些矩形块或特征点的运动)。

(3) 运动跟踪　从图像序列中估计物体的运动信息(参见**运动跟踪**)。运动跟踪关注的目标是物体,并维持物体的长时运动信息。

(4) 从运动恢复结构　基于运动跟踪或运动估计得到的刚体上一组点在图像序列中的二维运动轨迹,利用刚体约束估计物体的三维几何结构和摄像机运动参数(参见**从运动恢复结构**)。

(5) 运动图像压缩　基于运动跟踪或运动估计的结果进行运动补偿,用于运动图像压缩(参见**运动图像的压缩编码**)。

参考文献

1. Forsyth D A, Ponce J. 计算机视觉:一种现代方法. 林学訚, 王宏,等译. 北京:电子工业出版社, 2004

2. Sezan M I, Lagendijk R L, eds. Motion analysis and image sequence processing. Kluwer Academic Publishers, 1993

(邱慧军)

tuxiang yanse chuli

图像颜色处理(color based image processing)　借助颜色进行图像分析、分割、识别、合成等处理的方法、技术和应用。图像颜色是自然界中光及物体表面反射特性共同作用形成的一种视觉特征,被广泛应用于**数字图像处理**中物体的分割和识别,以及**计算机图形学**中**真实感图形生成**。由于人的视觉系统具有三种光感受细胞,机器视觉的传感器,如数码相机、数码摄像机中的光敏元件,一般都具有红、绿、蓝三种传感器单元,分别对可见光中的长、中、短波长的光敏感。这些传感器组成的阵列采集到的图像,即彩色数字图像,经过显示器或者印刷设备还原后,人们观察到的图像色彩和自然界的色彩非常接近。随着技术的发展,为了提高对光波长的分辨能力,有些机器视觉系统中有多于三种光传感器,如一些数码相机具有四种光传感器,以提高相机的颜色还原能力;多光谱和高光谱传感器对光的响应范围扩展到红外和紫外,对光波长的采样窗口多达数十至数百个,产生了多光谱和高光谱图像。这些彩色数字图像所表现出的颜色和自然环境,和人眼所见到的自然颜色有很大的不同。

总体来说图像颜色处理的应用可分为两大部分:①颜色的分析,其中包括颜色空间、颜色分割和颜色识别;②颜色生成,其中包括颜色合成和颜色映射。两者共同的基础是颜色模型,又名颜色空间或颜色系统。

颜色模型是对颜色的系统化的主观或定量描述体系,一般以颜色知觉为基础,对颜色进行分类排列或以三原色或四原色的相对混合数量来表示。由于各种研究和应用需求的不同,目前存在多种颜色模型,大体可以分为三类:①加色模型,如 RGB,YUV,HSL 以及 YCbCr 等,用于机器视觉、显示器等领域;②减色模型,包括 CYMK 及多色模型等,用于打印机和印刷工业领域;③认知模型,如孟塞尔颜色系

统、CIELAB 等,用于比色、配色和色差测量等。这些颜色模型之间,有些存在相互转换关系,有些则没有。

由于机器视觉的颜色来源于三种传感器,目前数字图像中颜色用红(R)、绿(G)、蓝(B)三维向量来表示,即 RGB 颜色空间。其中红、绿、蓝三种颜色称为三原色,其波长没有统一的严格定义,但是要求这三原色具有独立性,即三原色中的任意一种颜色都不能由其余两种颜色混合得到。RGB 颜色空间的主要问题是从 RGB 的坐标值不能直观地知道颜色,其中两个颜色向量之间的距离,也不能表示人所观察到的色差。为了解决这个问题,在**计算机视觉**、图形学、图像处理等实际应用中,提出了 HSL、HSV、HIS 等颜色空间,其中 H 是色调,S 是饱和度,L(Lightness)、V(Value)和 I(Intensity)都表示颜色的亮度。这些颜色空间根据颜色感知现象而建立,符合人们对颜色的直观感觉,即色调、饱和度、亮度,其坐标具有直观的颜色含义,广泛地用于计算机视觉和图像处理等领域,如基于颜色的目标检测、图像理解,以及医学图像分割等。

颜色分割是指基于颜色的图像分割,即分割图像中不同颜色的区域。这个看似简单的问题,如基于人的肤色在图像中分割人脸,基于天蓝色在图像中分割天空等,实际上却十分困难。首先是由于传感器的不同和人颜色认知中的颜色恒常性效应,人眼看上去很相近的颜色,如人的肤色、天空的蓝色、草地的绿色等,在图像中都包含一个相当大的色域。例如,肤色在实际分割中,难以和原木的颜色区分开来。

其次,颜色分割可以直接在 RGB 颜色空间,采用统计、聚类的方法进行分割,但常常需要考虑颜色空间的转换。如基于肤色的图像分割,则需要考虑肤色的特性。根据经验,我们知道:肤色是一种色调在红、黄之间,低饱和度的颜色;在一般的图片中,肤色的亮度中等。符合这些描述的颜色空间是 HSL、HSV、HSI 等颜色空间。因此,为了实现基于肤色的图像分割,需要先将 RGB 转化为 HSL,然后分割图像。

再次,物体的颜色容易受到物体的亮度和光照、阴影的影响。图像中最亮和最暗的物体,都是难以辨认颜色的,即颜色的饱和度很低。分割中容易出现比较大的误差和干扰。因此,基于颜色的图像分割,可以直接在 RGB 颜色空间中,采用统计、聚类等方法,得到需要分割的阈值,也可以考虑颜色的认知特性,将 RGB 值转换到 HSL 等颜色空间,再进行分割。无论采用哪一种方法,都容易受到光照环境的影响。

颜色识别实际是指颜色纹理识别,即根据彩色图像中某一区域的颜色变化规律,包括明暗、色调和饱和度的变化,来识别物体。基于颜色纹理的特征抽取和表示,相对来说,比灰度纹理更加复杂。对纹理的两大特性,自相似和重复结构,也更难描述。除了常用的统计分析法、信号分析法、结构分析法等方法以外,现代的颜色纹理识别还常用到稀疏表示、随机场等方法。

由于彩色图像本身的复杂性,图像中物体包含颜色、光照、视角等各种变化,一般难以得到完全精确的颜色纹理描述,因此,在实际应用中,需要综合运用**机器学习**和**模式识别**等多种方法,来实现对当前彩色图像纹理区域的识别。

参考文献

1. 陶霖密,徐光祐. 机器视觉中的颜色问题及应用. 科学通报,2001,46(3):178-190

2. 陶霖密,彭振云,徐光祐. 人体的肤色特征. 自动化学报,2001,12(7):1032-1041

3. 陶霖密,姚国正,汪云九. 人类颜色视觉的计算理论. 心理学报,1993,26(3):11-18

4. Jacobson E. Basic color: an interpretation of the Ostwald color system. Theobald Press, Chicago. 1948

(陶霖密)

tuxiang yundong mohu fuyuan

图像运动模糊复原(image motion-blurred restoration) 对由于运动而造成模糊的图像进行恢复的技术。运动模糊图像是由于照相曝光期间照相机与物体之间有相对运动而造成的。例如从飞机或卫星上拍摄地面照片或者用照相机拍摄高速运动物体的照片时,都有可能造成照片(图像)模糊。相对运动是均匀直线运动,这是经常遇到的典型情况。

假设清晰的图像是 $g(x,y)$,而造成 $g(x,y)$ 模糊的唯一因素是均匀直线运动所引起的位置移动,那么在曝光时间为 T 的情况下,可得模糊图像 $f(x,y)$ 的表达式如下

$$f(x,y) = \alpha \int_0^T g[x - at/T, y - bt/T]\,dt \quad (1)$$

这里假定在 T 时间内,照相机与物体之间的相对运动造成在 x 坐标方向移动了 a,y 方向移动了 b。式中 α 是和照片感光灵敏度有关的系数。

如果物体只是沿 x 坐标方向由左向右移动(或相机由右向左移动),将式(1)离散化并把比例系数 α 去掉(令 $\alpha=1$),则得到

$$f(x,y) = \sum_{k=0}^{\alpha-1} g(x-k,y) \quad (2)$$

式(2)表示模糊图像中的每一点 x 是由原始图像中的 x 到 $x-(a-1)$ 共 a 点叠加而成的。

当图像的宽度 $L=Ka$ 时(K 是整数),可以得到如下的近似解

$$g(x,y) \approx A - 1/K\sum_{k=0}^{k-1}\sum_{j=0}^{k} f'[x-ma+(k-j)a,y] + \sum_{j=0}^{m} f'(x-ja,y) \quad (3)$$

这里,$f'(x,y)$ 是 $f(x,y)$ 对 x 的差分,即

$$f'(x,y) = f(x,y) - f(x-1,y)$$

而 m 是一整数,取值应满足

$$x/a - 1 \leqslant m \leqslant x/a$$

式(3)中的 A 是一常数,当 K 值较大时,A 趋近 $g(x,y)$ 的平均值,在实际处理中 A 可以取 $f(x,y)$ 的平均值。

类似地,当照相机与物体之间的相对运动沿 y 方向移动了 b 时,复原图像是

$$g(x,y) \approx A - 1/K\sum_{k=0}^{k-1}\sum_{j=0}^{k} f'[x,y-nb+(k-j)b] + \sum_{j=0}^{n} f'(x,y-jb) \quad (4)$$

其中,$f'(x,y)$ 是 $f(x,y)$ 对 y 的差分,即

$$f'(x,y) = f(x,y) - f(x,y-1)$$

n 是一整数,取值应满足

$$y/b - 1 \leqslant n \leqslant y/b$$

根据式(3)和式(4),复原处理可推广到 x,y 方向都有移动的情况。图1给出了一个利用上述方法对运动模糊图像进行复原的例子。



(a) 模糊图像　　(b) 复原的图像

图1　均匀直线运动造成的模糊图像的复原

(此图取自参考文献3)

参考文献

1. 贾永红. 数字图像处理. 2版. 武汉:武汉大学出版社,2010

2. 赵荣椿,等. 数字图像处理导论. 2版. 西安:西北工业大学出版社,1995

3. Rafael C, Gonzalez/Paul Wintz. Digital image processing. 2nd ed. Addison-Wesley, 1987

4. Jain A K. Fundementals of digital image processing. Prentice-Hall Inc., 1989

(李树青)

tuxiang zengqiang

图像增强(image enhancement)　增强图像中所关心部分的信息,降低噪声,使图像更清晰或更易于分析的过程和技术。图像处理的基本运算中介绍的大部分技术都可以作为图像增强的基本技术。图像增强的目的有二:一是处理原始图像使它更适合于人的观察;二是变换图像以方便人或机器的分析和处理。增强过程不会增加图像的内在信息量,但却能增大所关心的信息的动态范围,使之更易于检测。用于图像增强的方法包括灰度变换(参见**图像点运算**)、图像平滑、图像锐化(参见**图像邻域运算**)、图像几何变换(参见**图像几何运算**)、图像伪彩色增强等。

在无法知道有关引起图像退化的定量信息的情况下,使用图像增强技术可较为主观地改善图像的质量。例如锐化技术可用于减少图像可能存在的模糊,而不管是何种原因产生的这种模糊。在物理本质上,运动模糊和聚焦不好产生的模糊是不同的,但相同的增强技术可能会改善任何一种模糊图像的质量。正因为增强技术并非是针对某种退化所采取的方法,所以很难预料哪些特定的技术对于给定的应用是最好的。因此图像增强算法一般是交互式的,并根据具体的应用来选择。尽管如此,图像增强仍是图像处理中一个非常重要的内容,因为它可以用于几乎所有的图像处理的应用中。

图像增强技术也经常用于改善图像中某种特征的可探测性,使人(或计算机)更容易观察到。减少图像模糊虽然更易于观察某些感兴趣的特征,但有时可能需要彻底改变图像的视觉效果以突出重要特征的可观察性,在这种情况下,可以把增强理解为增强感兴趣特征的可检测性,而不是改善图像的清晰度。例如,图像的边缘增强(锐化)处理也可以认为是图像增强,但其结果图像只突出了景物的轮廓、边界,而原图中一些平缓变化的细节都被忽视了。

伪彩色增强技术可将灰度图像的不同特征映射为不同的彩色,灰度图像映射成彩色图像。例如在广告中所用的图像颜色变换、二值化和轮廓效果,就是一反以往图像质量的逼真性要求,以得到一种特殊的艺术效果。

参考文献

1. Castleman K R. 数字图像处理. 朱志刚,等译. 北京:电子工业出版社,2002

2. Gonzalez R C,Woods R E. 数字图像处理. 2版. 阮秋琦,阮宇智,等译. 北京:电子工业出版社,2004

3. 许录平. 数字图像处理. 北京:科学出版社,2007

(朱志刚)

tuxing bianhuan

图形变换(graphics transformation) 将计算机生成的图形进行变换的过程和技术,是**计算机图形学**中较为基础的内容之一。通过变换可以从简单图形得到复杂图形;可以从某一个图形得到多个其他图形。广义上讲,图形变换包括模型变换、取景变换、投影变换、设备变换、视窗变换及**图形裁剪**。图形变换的核心问题是坐标系之间的转换。其中,模型变换是图形由局部坐标系到世界坐标系的变换;取景变换是图形由世界坐标系到观察者所在的照相机坐标系的变换;投影变换是将定义在三维空间中的图形转换到二维平面上的变换;设备变换将投影得到图形的二维齐次坐标表示转换到规格化设备坐标的变换,即将 x、y 坐标除以齐次坐标分量 ω;视窗变换是将定义在规格化设备坐标系中的图形转换到以屏幕像素为单位的图形的变换;图形裁剪则是计算出感兴趣的特定范围内的图形坐标问题。

参考文献

1. 唐泽圣,周嘉玉,李新友. 计算机图形学基础. 北京:清华大学出版社,1995

2. 唐荣锡,汪嘉业,彭群生,等. 计算机图形学教程. 修订版. 北京:科学出版社,2000

(周嘉玉)

tuxing caijian

图形裁剪(graphics clipping) 将落在窗口或观察空间内的图形映射到视图区中,而将落在外面的图形裁剪掉的过程和技术。窗口一般是矩形或长方体,用窗口对平面图形作裁剪称为二维裁剪,这是图形裁剪的基础。如图 1 所示,设某待裁剪线段的起始点和终点坐标分别为(A,B)和(C,D),窗口的四条边为 x_L,x_R,y_B,y_T,则线段裁剪后的起始点和终点坐标分别为(A',B)和(C',D)。

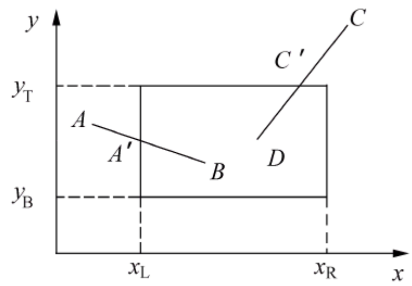


图 1 图形裁剪

裁剪包含两部分内容:①点在区域内、外的判断;②计算图形元素与区域边界的交点。由于交点计算比较费时,对于比较复杂的图形,为避免不必要的计算,可先做图形的最大最小包围盒与裁剪窗口的重叠测试。裁剪算法效率的高低直接影响整个图形系统的效率,要根据实际情况选择不同的裁剪方法。裁剪包括二维线段裁剪、多边形裁剪和字符裁剪等。

参考文献

1. 唐泽圣,周嘉玉,李新友. 计算机图形学基础. 北京:清华大学出版社,1995

2. 唐荣锡,汪嘉业,彭群生,等. 计算机图形学教程. 修订版. 北京:科学出版社,2000

(周嘉玉 金小刚)

tuxing chuliqi

图形处理器(graphic processing unit,GPU) 面向图像运算工作的可编程**微处理器**。图形处理器广泛应用于嵌入式设备、移动设备、游戏机、个人电脑、工作站等各类设备或计算机中。现代的图形处理器通常基于高度并行的、且对图形处理运算做了定制设计的内部硬件结构,主要包括多边形转换、光源处理以及视频处理功能等,从而能够比中央处理器更为有效地进行图形处理,图形处理器的出现减轻了中央处理器的处理负担。图形处理器可以集成在显卡上,或者直接集成到计算机主板上。

GPU 是 NVIDIA 公司在发布 GeForce 256 绘图处理芯片时首先提出的概念,在此之前,计算机中处理图像输出的显示芯片,通常很少被视为是一个独立的运算单元,但这设计理念则由来已久。

20 世纪 70 年代,ANTIC 和 CTIA 芯片为 Atari

游戏机(一种早期的黑白游戏机)提供了硬件控制的图形和文字混合显示模式以及其他视频效果的支持,ANTIC 芯片是一个特殊用途的处理器,用于映射文字和图形数据到视频输出。

80 年代,IBM 推出了第一个用于个人计算机 2D/3D 图形加速处理的控制器,比硬件 3D 加速成为个人计算机的标准配置的时间早了整整 10 年。但因高价格、较低的运行频率以及软件兼容性问题,其没有获得商业上的成功。而 Commodore Amiga 是第一个在市场上获得成功的在其视频硬件上包含 2D 图形位图处理功能的计算机,也是世界上首款多媒体、多重任务处理的计算机。Commodore Amiga 的独特之处在于,其实现了一个完整的具有独立指令集的图形处理加速器,将线条绘制、区域填充、图形转换等功能都从中央处理器中剥离出来。

90 年代,Microsoft Windows 的崛起引发人们对高性能、高分辨率 2D 位图图形运算的兴趣。在个人计算机市场上,Windows 的优势地位意味着 PC 图形厂商可以集中精力发展单一的编程接口,即图形设备接口(graphic device interface)。1991 年,S3 Graphics 推出了第一款单芯片的 2D 图像加速器,名为 S3 86C911。其后,86C911 催生了大量的仿效者,到 1995 年,所有主要的 PC 图形处理器制造商都于他们的芯片内增加 2D 加速的支持。此时,固定功能的 Windows 图形加速处理器的性能已超过昂贵的通用图形协处理器,令这些协处理器逐渐从个人计算机市场上消失。在整个 90 年代,2D 图形持续加速发展。随着制造能力的改善,图形处理器的集成水平也在提高。同时,能够应用于多个图形处理领域的应用开发接口也陆续出现,包括供微软 Windows 3.x 使用的 WinG 图像程序库及其后继 DirectDraw 接口。

在 20 世纪 90 年代初期和中期,CPU 辅助的实时三维图像处理越来越常见于个人计算机和电视游戏上,从而产生了大量的硬件加速的 3D 图像处理需求。早期出现于大众市场上的 3D 图像硬件处理的例子有第五代视频游戏机,包括 PlayStation 和任天堂 64。而在个人计算机领域,最初始的低成本 3D 图形处理器的生产尝试都失败了,包括 S3 的 ViRGE、ATI 的 3D Rage 和 Matrox 的 Mystique。这些芯片的特点是在上一代的 2D 加速器上加入 3D 功能,有些芯片为了便于制造和降低成本,甚至使用与前代兼容的针脚。后来,出现了专门用作 3D 图形加速处理的独立显卡(没有 2D 加速功能),如 3dfx 的 Voodoo。逐步地,随着芯片制造技术的进展,视频、2D、3D 加速处理功能都集成到一块芯片上,Rendition 的 Verite 是第一个能做到这样的芯片组。

OpenGL 是出现于 20 世纪 90 年代初的专业图像编程接口,并成为了个人计算机领域图像处理发展的主导力量以及促进硬件发展的动力。微软的 DirectX 在 90 年代末开始受到 Windows 游戏开发商的欢迎,不同于 OpenGL,微软坚持提供严格的一对一硬件支持。但许多的 GPU 生产厂商都提供了自己独特的功能,使得这种做法并不受到欢迎;而当时的 OpenGL 应用程序已经能满足 GPU 生产厂商的需求,导致 DirectX 往往落后于 OpenGL 一代。逐步地,微软开始与硬件开发商有更紧密的合作,并发布支持特定图形硬件加速功能的 DirectX。Direct3D 5.0 是第一个增长迅速的此类编程接口版本,在游戏市场中获得迅速普及。Direct3D 7.0 支持硬件加速多边形转换和光源处理,这样,3D 图形加速处理器发展到了另一个重要的阶段,即支持 3D 渲染流水线的阶段。NVIDIA 的 GeForce 256 是第一个在市场上获得成功的此类显卡。硬件加速多边形转换和光源处理的出现,为其后更为灵活和可编程的硬件像素着色引擎和顶点着色引擎的出现铺平了道路。

21 世纪以来。随着 OpenGL API 和 DirectX 类似功能的出现,GPU 增加了可编程着色的能力——每个像素可以经由独立的小程序段处理,当中可以包含额外的图像纹理输入,而每个几何顶点同样可以在投影到屏幕上之前被独立的小程序段处理。NVIDIA 是首家能生产支持可编程着色芯片的公司,即 GeForce 3。2002 年 10 月,ATI 发表了 Radeon 9700,它是世界上首个 Direct3D 9.0 图形加速处理器,而像素和顶点着色引擎可以运行循环和长时间的浮点运算,这就使其编程更为灵活,并达到更快的运算效果。随着处理能力的增加,GPU 的功耗也随之增加,高性能 GPU 往往比目前的中央处理器消耗更多的电能。

硬件结构

现代的图形处理器通常集成多个高速图形处理单元,这些单元也往往分为多类以适用于不同的图形处理功能,如纹理处理、光源处理、坐标处理等。这些单元之间由高速内部总线或互联交换网络相连构成流水段结构,以提高图形处理的吞吐率。GPU 内部通常集成一定数量的内存,以便存储计算过程中的中间结果,减少与外部的数据交换。同时,现代 GPU 通常支持其自有的指令集,并采用 SIMD 编程模式。

发展趋势

图形处理器的一个重要发展趋势是 GPGPU (general purpose graphics processing unit)的应用与蓬勃发展,主要出发点就是将现代图形处理器的高性能可编程浮点计算能力应用于通用计算,而不仅仅是图形处理相关的任务。GPGPU 已成功应用于代数计算、流体模拟、数据库应用、频谱分析等非图形应用领域。与 CPU 相比较,CPU 中的大部分晶体管主要用于构建控制电路(如分支预测等)和 Cache,而完成实际的运算工作的资源则不多。而 GPU 与 CPU 的设计目标不同,其控制电路相对简单,而且对 Cache 的需求较小,所以大部分晶体管可以组成各类专用电路和多条流水线,使 GPU 的计算速度有了突破性的飞跃,拥有惊人的处理浮点运算的能力。

GPU 在高性能计算方面所具有的优势(或特点)主要包括:高效的并行性、高密度的运算能力以及超长图形流水线。因此,其在大规模的数据流并行处理方面具有明显的优势。

参考文献

1. Akenine-Moller T, Haines E, Hoffman N. Real-time rendering. 3rd ed. Wellesley: A K Peters, 2008

2. 张舒,褚艳利,李开勇,等. GPU 高性能运算之 CUDA. 北京:中国水利水电出版社, 2009

3. 林一松,唐玉华,唐滔,等. GPGPU 技术研究与发展. 计算机工程与科学,2011, 10(33):85-92

(张悠慧)

tuxing fanzouyang jishu

图形反走样技术(anti-aliasing technique) 消除在计算机图形生成中因离散化而引起的误差的技术,又称反混淆技术。计算机图形是由离散点组成的数字化图像,因而每幅图像都是对连续色彩真实画面的一个采样。因此,计算机图形必然与真实景物之间存在误差。这种误差表现为图形上的直线或曲线呈现锯齿状,色彩组成及纹理引起失真,细小物体在画面上得不到反映等。这些问题在**计算机图形学**中称为图形走样或混淆。图形走样在直线和多边形生成、图形线/面消隐、阴影生成(参见**光线跟踪技术**)、纹理生成、光线跟踪和辐射度绘制(参见**辐射度技术**)等情况下都可能以各种不同的方式存在。

从信号理论的角度来看,在图形生成中,由于使用采样技术,当采样频率太低时便会导致图形走样。解决图形走样问题的最根本方法是提高采样频率或者以面积采样代替点采样。提高采样频率的常用办法是提高分辨率,即高分辨率计算,低分辨率(像素分辨率)显示。其实现方法是将要显示的像素划分成许多子像素,然后按通常算法计算出各个子像素的颜色或灰度值,最后将所有子像素的颜色平均值作为像素的显示值。该方法算法简单,但所需要的存储和计算量较大。为了减少计算量,常常使用非均匀地增加采样频率的方法,例如只在图形细节较强或较复杂的部分增加采样频率。采样点的分布也可以是不规则的,这在绘制许多复杂场景时尤其有效。特别是在利用光线跟踪技术绘制诸如毛绒表面、分数维几何面、几何纹理面等复杂场景时,采用随机分布(如蒙特卡罗分布)的采样对于反混淆至关重要。事实上,辐射度技术中采用的层次辐射度技术对于几何面片划分的非均匀采样亦起到了反走样的效果。

用面采样代替点采样对于反走样是行之有效的。通常在图形绘制时,为了使算法简单都把像素作为一个点来处理。实际上,如果将它看成是一块极小的屏幕区域,将大大减少图形走样现象。此时像素区域与各种图元如直线、多边形相交时,将严格计算相交的面积,其像素的最终颜色值将由与之相交的各种图元的颜色值,利用其相交面积加权平均获得。该方法可成功地用于直线和多边形生成、消隐等的图形反走样。

参考文献

1. 唐荣锡,汪嘉业,彭群生,汪国昭,等. 计算机图形学教程. 修订版. 北京:科学出版社,2000

2. Foley J D, Dam A V. 交互式计算机图形学基础. 唐泽圣,周嘉玉,等译. 北京:清华大学出版社,1986

(吴恩华)

tuxing shipeiqi

图形适配器(graphics adapter) 一种专用于图形处理与输出的计算机扩展卡。通常提供二维或三维图形渲染、视频解码、电视信号输出、多显示器支持等功能,简称图形卡或显卡。独立的图形适配器(简称独显)往往通过计算机主板上的 PCI、AGP、PCI-E 等总线插槽与主机连接,并集成有单独的显示内存,加速二维及三维图形处理和绘制的性能较高;集成的图形适配器(简称集显)则集成在主板上,且共享系统主存作为显示内存,三维处理等性能较低,属于低端适配器。

最早的彩色图形适配器(color graphics adapter,

CGA)是IBM公司在1981年开发的,它也是第一个IBM PC上的计算机显示标准。该适配器具有16K字节显示内存,提供多种图形和文字显示模式,可以达到640×200的显示分辨率以及最高16色的显示能力。增强图形适配器(enhanced graphics adapter,EGA)是IBM公司在1984年为其PC-AT计算机引入的,可以在640×350的分辨率下显示16色。EGA包含16K字节的只读存储器来扩展系统BIOS以便实现附加的显示功能。1987年,IBM公司推出了称为视频图形阵列(video graphics array,VGA)的显示标准及对应的图形适配器,支持640×480的分辨率及256种颜色(这256种颜色可以在总共262 144种颜色中挑选出来)。此后,随着图形处理器功能的日益增强,图形适配器所能支持的分辨率越来越高,显示颜色也越来越丰富,此外,还逐步集成了二维及三维图形渲染、视频解码、电视信号输出等功能,逐步将CPU从图形处理以及相关的运算中解脱出来。

图形适配器主要由以下5个部分构成:

(1)图形处理器　一种专用**微处理器**,主要用于在图形处理流程中分担一部分中央处理器(CPU)运算功能,加速二维以及三维图形的处理和绘制。它具有高度并行的结构,在执行图形处理的复杂算法时,有着比通用CPU更高的效率(参见**图形处理器**)。

(2)视频基本输入输出系统(BIOS)　即集成于适配器上的固件,一般实现了控制适配器基本操作的功能,并向主机系统提供一系列的操作接口,便于后者操控该适配器进行图形处理。通常,视频BIOS还包含有**图形处理器**的内存操作时序、操作频率、电压等信息以及显示内存等其他信息。

(3)显示内存　集成于适配器上的高速或多端口内存,一般容量在几兆到几千兆字节。该内存用于存储当前在显示器上显示的图像数据以及图形处理过程中所需的中间数据,如顶点数据缓存、纹理数据以及三维图形处理中的深度缓冲数据。(参见**视频图形随机存取存储器芯片**)。

(4)基于随机存取器(RAM)的数模转换器(RAMDAC)　通常用于需要模拟输入信号的**显示器**,如彩色阴极射线管(CRT)显示器。随着数字信号显示器(如液晶显示管(LCD)显示器)的日益普及,RAMDAC已不再作为一个独立的部件存在。

(5)输出接口　用于连接图形适配器与显示器的接口部件,常见的有以下几种:用于模拟信号输出的VGA接口,用于连接高分辨率LCD显示器的DVI(digital visual interface)接口,用于连接高清电视的HDMI(high-definition multimedia interface)接口。

随着图形处理器性能的不断提升,图形适配器的三维图形处理能力持续增长,对配套显存容量的需求也不断提高。除了用于图形处理,有越来越多的应用利用图形适配器的数据并行计算能力,辅助CPU完成一些通用的计算任务。

参考文献

1. Luna F D. Introduction to 3D game programming with DirectX 10. Jones & Bartlett Publishers, 2008

2. 赵中秋. 显卡维修知识精解. 北京:电子工业出版社,2010

(张悠慧)

tuyuan shengcheng

图元生成(graphics primitive generation)

根据几何图元参数在图形输出设备上生成图元的过程和技术。通常把构成图形的基本元素称为**图元**,一般包括点、直线、曲线、圆、多边形、位图和字符等。研究这些图元的生成算法是**计算机图形学**的基本内容之一。

在图形输出设备上输出一个点时,需要把该点的坐标信息转换成控制输出设备的相应指令。以阴极射线管监视器为例,需要在指定的屏幕位置上开启(接通)电子束使该位置上的荧光点发亮,因此应将帧缓存中相应坐标位置的内容置为“1”。当电子束“扫视”每一条水平扫描线时,一旦遇到帧缓存中值为“1”的点,就发射一亮光,即输出一个点。

在图形设备上输出一条直线时,需在应用程序中给出该直线端点的坐标,然后由输出设备将这对端点间的路径进行绘制。由于大多数图形设备都只提供x方向和y方向动作的信号,所以只能准确地画出水平或垂直直线。而对于任意斜率的直线,就必须由算法来决定图形设备何时以及应往何方向动作。衡量直线生成算法的优劣,一是看生成的直线是否很好地逼近所需的直线,二是看生成直线的速度如何。由于光栅扫描设备各像素点的坐标是用整数表示的,因此,对直线各轨迹点坐标值的实型数,需作四舍五入,近似成整数值,致使所生成的直线出现台阶或锯齿状。为改进直线的质量,可以采用高分辨率的显示设备,也可采用反走样技术。

利用点和直线的生成算法,也可以生成曲线、圆、多边形、字符等图元。但是,由于这些图元使用频繁,各有特点,通常仍分别研究其快速生成算法。大部分图形加速卡支持用硬件实现点、直线、曲线、

圆、位图和字符的绘制,用户只需调用 OpenGL 和 DirectX 中相应的函数即可。

参考文献

1. 唐泽圣,周嘉玉,李新友. 计算机图形学基础. 北京:清华大学出版社,1995

2. 唐荣锡,汪嘉业,彭群生,等. 计算机图形学教程. 修订版. 北京:科学出版社,2000

(周嘉玉 金小刚)

tuandui ruanjian guocheng moxing

团队软件过程模型(team software process model) 为软件开发团队的建立和管理提供过程指导和实践的工作框架。于 2000 年由美国卡内基梅隆大学的软件工程研究所(SEI)提出,简称 TSP。软件开发需要团队的合作,一个好的开发团队不仅仅是一组人在一起工作,还要求整个团队可以在预计的进度计划范围内开发高质量的产品。TSP 的目的就是提供了一个定义良好的、可操作的过程,帮助团队建立一致的目标、共同的过程、获得可以覆盖任务要求的技能和经验、实现有效的领导、明确的角色和责任分配、良好的沟通、透明的进展状态了解以及保持持续的训练。TSP 的过程如图 1 所示。

TSP 过程的第一步是团队建立过程,期间所有团队成员要制定战略、过程和计划,然后遵循他们自己定义的过程进行开发工作;此后,团队要定期评审和调整策略、目标和计划,迭代地进行开发。

TSP 要求参与开发团队的成员,应该首先接受个人软件过程(PSP)的训练,TSP 的结构如图 2 所示。

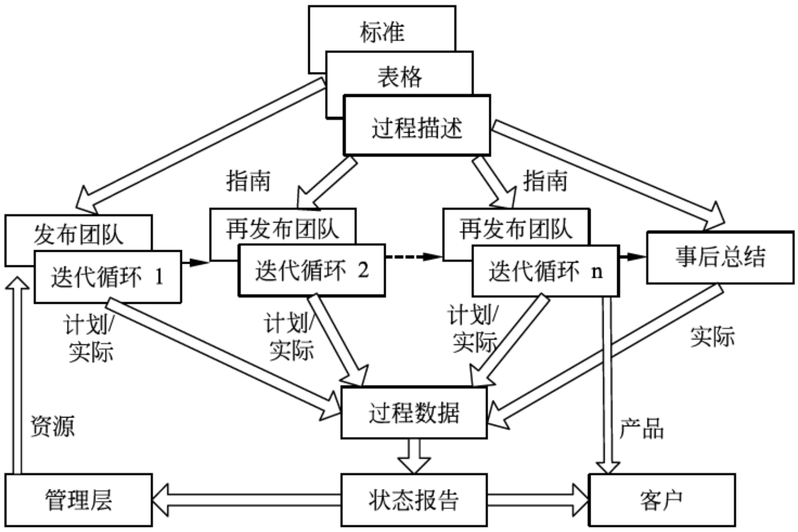


图 1 TSP 的过程流

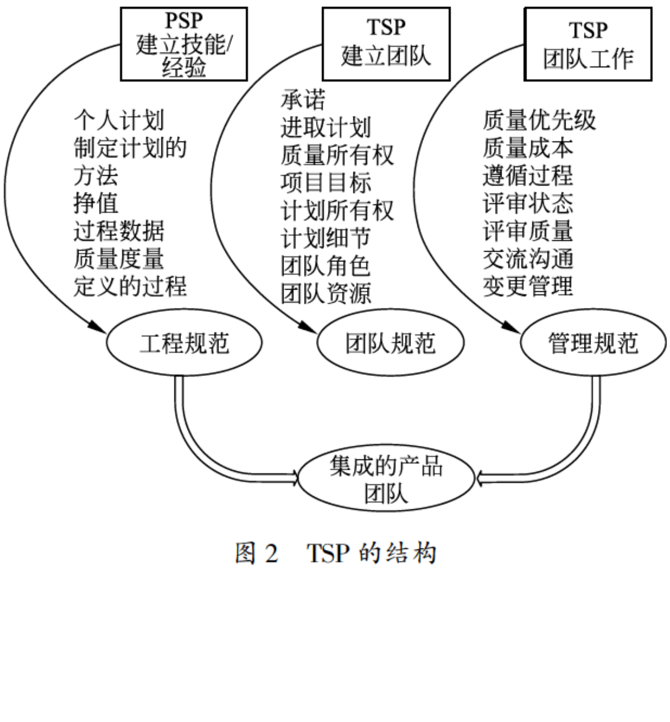


图 2 TSP 的结构

参考文献

1. Humphrey W S. TSP^SM: Leading a development team. Addison Wesley, 2006

2. Humphrey W S. TSP^SM: Coaching development team. Addison Wesley, 2006

（王青）

tuiliji

推理机（inference engine） 一个或一组计算机推理程序，是知识系统中实现基于知识推理的部件。基于知识的推理是指依据一定的规则，运用知识从已有事实导出结论的过程。推理机用来控制和协调**知识库**与数据基的运行，利用知识库中的知识，按一定的控制策略求解问题。

知识的选择和运用原则称为**推理控制策略**。一般来说，知识系统中包含大量知识，因此有效选择与运用知识就成了推理机的关键任务。推理控制策略对这一任务的完成起了重要作用，其能否迅速准确地找到与最终解有关的知识，直接影响推理的效果与效率。

产生式表示是一种常用的**知识表示**形式，产生式表示亦称规则或产生式规则。若产生式系统（或曰基于规则的系统）中以产生式来表达领域知识，则这样的产生式系统是一种基于知识的系统（简称知识系统）。

下面以产生式系统为例，对推理机（亦称推理程序）做进一步介绍。推理机是产生式系统的核心，其性能优劣对产生式系统性能有重要影响。推理机负责产生式规则前提条件的测试和匹配，规则的调度和选择，动态数据基的状态更新，匹配冲突消解和推理结束判断等。推理机按一定策略从规则库中选择规则与数据基中的事实进行匹配，当匹配成功的规则多于一条时，需根据一定的策略消解冲突，从中选出一条本次迭代中被执行的规则。

基于典型推理控制策略的推理控制包括正向推理（也称数据驱动的控制、自底向上的控制），反向推理（也称目标驱动的控制、自顶向下的控制），以及双向推理（也称混合控制）。

正向推理 从初始数据出发，正向使用规则，导出结论的推理。其推理过程为：系统根据用户提供的初始信息（数据），在知识库中寻找能与之匹配的规则，若能找到，则将被选中规则的结论部分（⊄数据基）添加到数据基中，重复这一过程，直至得出答案或不能求解而终止。

正向推理简单、易实现，但目的性不强，有时效率较低。此外，如果只能进行正向推理，系统的解释功能会受到影响。

反向推理 提出假设或目标，将目标与数据基中的数据或产生式的结论部分匹配。与数据匹配成功则目标被验证；与某产生式结论部分匹配成功，则该产生式前提条件成为新目标。重复这一过程，直至初始假设（或目标）为真或为假。若初始目标为假，则再提出新的初始假设或目标。

显然，反向推理在选择目标时常常呈现盲目性，因此，反向推理比较适合结论单一或直接提出结论要求系统证实的情况。

双向推理 依据初始数据借助正向推理帮助提出假设，再用反向推理进一步寻找支持假设的证据，反复这个过程，直至获得满意的结论。

双向推理集中了正向和反向推理的优点，但其控制策略较前两者复杂。

参考文献

1. 陆汝钤. 人工智能（上）. 北京：科学出版社，1989

2. 王士同. 人工智能教程. 北京：电子工业出版社，2001

3. Giarratano Joseph, Riley Gray. Expert systems: Principles and programming. 4th ed. Boston: PWS Publishing Company, 2004

（刘大有 虞强源）

tuoji shouxie hanzi shibie

脱机手写汉字识别（off-line handwritten Chinese character recognition） 通过扫描仪将稿纸上手写的**汉字**输入计算机并进行**汉字识别**的过程和技术。

由于不同书写者书写汉字的巨大差异和变化，再加上无法直接获得书写过程的笔画信息，使脱机手写汉字识别成为最困难的汉字识别问题。手写汉字与标准楷书的差异和变化程度，将直接影响到识别的难度。据此，将手写汉字识别粗略分为规则书写汉字识别与自由书写汉字识别。规则书写汉字是指书写的汉字基本保持横平竖直笔画的手写汉字，而自由书写汉字则包括自由连笔、笔画弯曲的草书书写汉字。

由于手写汉字的巨大不确定性，人们对于识别脱机汉字的方法在相当长时间内采取结构分析的方法，即从汉字图像中抽取汉字笔画，按照笔画的属性和逻辑关系构造汉字的结构属性描述，进行结构匹配识别。但是实践证明，这种方法一直未能取得成效。

在印刷汉字识别（参见**印刷体汉字识别**）取得成功的启发下，人们自然地将基于汉字全局信息的统计模式识别方法应用于手写汉字识别。因为，手写汉字识别和印刷汉字识别最主要的差别在于，手写汉字的类内变化要大大超过印刷汉字，仅此而已。这一思路的转变，使脱机手写汉字识别的研究取得进展。采取从基元的笔画微结构出发（如四方向线素特征等），利用统计的全局信息进行的统计判决识别，提供了脱机手写汉字识别的基础。

为了较好解决脱机手写汉字识别问题，不仅在字符图像的非线性规一化、特征提取、鉴别特征选择优化等方面做了进一步的研究和改进，而且在分类器设计、鉴别学习算法、支持向量机（SVM）分类诸方面都有相当的发展，促进了脱机手写汉字识别问题的进一步解决。

目前，手写规则汉字的识别率已达 98% 以上。自由书写手写汉字的识别率也可达 90% 以上。

关于自由书写手写汉字识别研究的困难，与其说是在算法上，还不如说是在样本的收集上。因为，自由书写手写汉字的随意性，造成字形的随意变化，对于人都难以辨识的汉字，要求计算机不经过学习就能识别是不现实的。应当说，在样本足够的条件下，解决自由书写手写汉字的识别应当是可以实现的。

参考文献

1. 郭平欣，张淞芝．汉字信息处理技术．北京：国防工业出版社，1985

2. 张炘中．汉字识别技术．北京：清华大学出版社，1992

3. 丁晓青，王言伟，等．文字识别：原理、方法和实践．北京：清华大学出版社，2016

（丁晓青）

W

waibu luyou xieyi

外部路由协议(exterior routing protocol) 在大型网络中自治系统(AS)间使用的**路由协议**。大型的复杂网络往往由许多自治系统组成(例如,互联网由数万个AS组成),每个自治系统有独立的管理结构,可运行自己的**内部路由协议**,可执行特殊的路由策略和商业模式。一个自治系统通常包括多个路由器,路由器之间遵守预定的路由协议交换路由信息,完成报文分组从源和目标的传送。连接属于不同自治域的路由器之间运行的路由协议为外部路由协议。自治系统有时也称为域,故外部路由协议也叫**域间路由协议**。自治系统用AS标识码标识,随着自治系统数量的急剧增加,16位的AS标识码已扩充为32位。

内部路由协议执行源IP地址到目的IP地址之间的路由,主要关心如何选择具有最优开销费用的路由。外部路由协议执行源AS到目的AS之间的路由。由于外部路由协议在多个不同自治系统之间运行,每个自治系统的路由策略、安全策略和商业模式各异,因而它不仅关心如何找到最佳或最经济的路由,而且还要根据路由策略来操纵路由,并对整个互联网的扩展性、稳定性、安全性和性能优化等产生影响。例如,不同路由策略可能引发路由震荡(路由器之间不停交换大量路由信息,路由器反复计算路由,网络系统无法稳定)、路径增长(源和目标的传送路径大大超出最短路径)等问题。再例如,某个国家的报文不希望在传送的过程中通过其敌对国家中的自治系统,某个公司的报文在传送过程中不希望通过某潜在竞争对手的自治系统,因而产生安全机制和信任机制问题。因此,外部路由协议远比内部路由协议复杂,是互联网研究的热点之一。

早期互联网采用的外部路由协议是外部网关协议(EGP),为了克服其缺点,在1995年提出并规范了**边界网关协议**(BGP-4)。随着设备制造商对BGP功能的扩展和完善以及IETF的域间路由(IDR)工作组多年来的努力,BGP-4已成了互联网最成熟的协议之一。与BGP相关的RFC规范有60多个。

参考文献

谢希仁. 计算机网络. 5版. 北京:电子工业出版社, 2008

(龚正虎)

wai cunchu shebei jiekou

外存储设备接口(external storage device interface) 计算机主机与外存储器之间的交接部分。

外存储设备接口的基本功能如下:

(1) 地址译码 一个计算机系统通常带有许多种外围设备,通过接口地址对外围设备进行选择。

(2) 信息交换 通过接口,计算机把数据和控制信息送给外围设备,外围设备把数据和状态信息送到计算机,即通过接口实现数据的输入和输出以及控制信息的输出和状态信息的输入。

(3) 数据格式的转换 计算机与各种类型的外围设备在编码、数据格式等方面是不相同的。因此,接口具有数据的分解或组装、串并行格式的转换、不同编码方式之间相互转换等功能。

计算机要做到部件的通用性,要实现可靠的数据交换,必须对部件进行标准化,约定好各种类型的外围设备接受的指令形式和状态控制信息以及通信的方式。这种约定就是接口标准或规范。也就是说,接口除了电缆、电路芯片、接插座这些硬件之外,还需要管理程序之类的软件。

计算机外存储器有两种接口,即系统级接口和设备级接口。早期外存储器多采用设备级接口,随着技术的进步,现在多采用系统级接口。

设备级接口目前主要有IDE、EIDE、USB、SATA、并行SCSI和SAS协议,其中USB通常用于直接连接便携式/**移动存储器**,SATA和SAS协议是在并行ATA和SCSI的基础上串行化接口协议,一般只用于直接连接**硬磁盘驱动器**,而并行SCSI协议除了用于直接连接硬磁盘驱动器外,也会用在一些中低端**磁**

盘阵列设备中，提供到主机的连接。

系统级接口主要用于将存储节点连接到服务器上，或者用于将多存储节点和多服务器连接构成局域或广域存储区域网，如 FC、iSCSI 和 IB。其中，在某些高端存储设备中，也有直接使用系统级接口（如 FC 等）连接硬磁盘驱动器的。

在计算机中常用的外存储设备有**硬磁盘驱动器**、**软磁盘驱动器**、光碟驱动器、磁带驱动器（参见**磁带机**）等，下面分别介绍它们的接口。

1. 硬磁盘驱动器接口　硬磁盘驱动器是计算机最重要的，也是发展最快的外存储设备。早期采用 SMD，ST 506/412，ESDI 、IDE、EIDE 等接口，现在多采用并行 SCSI、SATA 和 SAS 接口，一些高端存储设备也会用到 FC、IB 等。

从接口的串并性特点看，早期以采用并行接口为主。但像 EIDE，SCSI 这一类并行接口有其固有的缺点，进一步提高数据传输速率有一定的难度。2000 年以来，各种串行接口相继问世，如 USB，IEEE 1394，SATA，SAS 以及 iSCSI 等，在外围设备中应用越来越多（参见**输入输出设备接口**）。

IDE 接口（ATA 接口）　是 1984 年 Compaq 公司和 Western Digital 公司提出的。它的特点是将控制器与驱动器做成一体。考虑到当时微型机主要采用 AT 总线，IDE 接口中许多信号直接取自 AT 总线，原封不动地送往硬磁盘机。适配器只需要做简单的总线缓存和地址转换，因此适配器的结构也非常简单，安装连接简便，价格也较低。由于它采用了 AT 总线信号，也被称为 ATA 接口。为了和后来开发的 SATA 相区别，有时也称之为 PATA 接口。

ST506/412 接口规定，硬磁盘驱动器每道为 17 个扇区。由于 IDE 接口的硬磁盘机将控制器和驱动器做在一起，每道扇区数可根据存储密度选择，记录方式也可选择，极大地提高了整机容量。而扇区数和记录格式的不同通过控制器自动转换成 AT BIOS 能支持的驱动器类型。IDE 硬磁盘机内部都有数据缓冲寄存器，读写的数据通过接口时采用 8 位并行传送，数据传输速率要比 ST506/412 高得多，采用 PIO 模式，可达到 3.3 MB/s。

IDE 接口理论上能管理的容量可达 137 GB，但实际上受基本输入输出系统（BIOS）的限制，只能达到 528 MB。

IDE 接口硬磁盘机与适配器之间的连接用一根 40 线的电缆，长度限制为 457.2 mm（18 in）。每根电缆可接主从两台硬磁盘机。

EIDE 接口　对 IDE 进行了改进。IDE 接口的最大缺点是其管理的硬磁盘机容量不能超过 528 MB，这显然不能满足硬盘单机容量日益增长的要求。EIDE 与 IDE 接口完全兼容，与适配器也是通过 40 线的扁平电缆相连，插针信号定义也一样。

EIDE 接口是硬磁盘机常用的接口之一。EIDE 接口有许多规范，如：Fast ATA，Ultra ATA，Ultra DMA 等。它们之间的差别在于有不同的数据传输模式和数据传输速率。

并行 SCSI 接口　处于主机适配器与智能控制器之间的一种系统级并行接口。SCSI 总线通过适配器与主机相连，通过各种智能控制器和外围设备相连。主要特点如下：

（1）作为一种通用接口，可连接各种采用 SCSI 接口的外围设备，如硬磁盘机、软磁盘机、磁带机、打印机、光碟机、扫描仪、网络设备等。

（2）系统的配置灵活。在单一总线上一台外围设备能与多台主机进行通信，一台主机也能与多台外围设备进行通信。构成单主机多控制器系统时，总线上可连接 8 个设备。设备可以是 SCSI 接口的外围设备，也可以是控制器，但其中有一台必须是主适配器。如果有设备是控制器，每台控制器还可以带 7 台设备。利用 SCSI 总线还可以组成多主机多控制器系统，这时总线上的设备数仍是 8 个。总线上的设备分为启动设备和目标设备，启动设备是发出命令的设备，目标设备是接受并执行命令的设备。每一种设备可以是启动设备也可以是目标设备，总线上的所有设备都能相互通信。

（3）对硬磁盘机地址的管理，采用连续的数来编排，也就是只有扇区号。对于硬磁盘机具体的物理参数，有几个柱面、几个磁头或每道有几个扇区，在编程时都不必考虑。

SCSI-1 接口采用 8 根数据线和 1 根奇偶检验线并行传送数据，还有 9 根控制和状态信号线。电缆采用差分驱动和单端驱动两种方式。单端驱动采用一个信号一根导线，所有信号的地线公用一根导线。差分驱动则每个信号都有自己的返回线，要用两个导线（通常采用双绞线）。上述电缆称为 A 电缆，标准规定采用 50 针接插座。SCSI-2 采用 16 位和 32 位并行，定义了采用 68 线的 B 电缆。

SCSI 接口曾广泛应用于各种高档微机、工作站、服务器。硬磁盘机、只读光碟机、磁带机和扫描仪等外围设备都有 SCSI 接口的产品。

1986 年美国国家标准协会公布了 SCSI 标准,即通称的 SCSI-1 标准。后来,SCSI 接口也有许多规范,如 SCSI-2,SCSI-3,Fast SCSI,Wide SCSI,Ultra SCSI,Wide Ultra SCSI 等。这些规范有不同的总线宽度和数据传输速率。2002 年发表的 Ultra 320 SCSI 采用 16 位并行传送,数据传输速率为 320 MB/s。

SATA 接口 2001 年秋提出的接口规范 SATA(serial advanced technology attachment)即串行 ATA,是新一代的 ATA 内部存储接口,目的是克服 ATA(EIDE)的缺点。它将数据并行传送改为串行传送,使用 4 列信号线,即电源、地线、发送数据线和接收数据线,与 ATA 相比要简单多了。SATA 接口对数据和命令都有纠错能力,数据传输电压用 250 mV。采用点对点连接,每个通道只接一个设备,数据传输速率可达到 1.5 Gb/s。同并行 ATA 相比,SATA 有以下优点:

(1) 为未来提供更方便的扩展功能,提供从 1.5~3.0 Gb/s 甚至更高的传输率。

(2) 使用更细更灵活的电缆,从而更易于集成以及改善系统内部的空气流通。

(3) 使用更少的信号导线,电路板层得到了极大的改善。

(4) 规范同时包括额外的像热插拔、原生命令队列和封装管理等增强功能。

自 2001 年正式确立 SATA 1.0 后,至 2007 年 3 月一共有 SATA1.0,SATA1.0a,SATA2.0,SATA2.5 以及 SATA2.6 等五个规范推出。

SAS 接口 2001 年 11 月 26 日,Compaq、IBM、LSI Logic、Maxtor 和 Seagate 宣布成立 Serial Attached SCSI 工作组,目标是将并行 SCSI 与 SATA 的优点相结合,定义一个新型串行点对点的企业级存储设备接口。2003 年 5 月,SAS 1.0 规范正式出台并提交给 ANSI(美国国家标准协会)讨论,同年 9 月,SAS 1.0 正式通过 ANSI 认证。2003 年 7 月 10 日,T 10 公布了 SAS 1.1 工作草案的第一个版本(00 修订本),它的绝大部分内容与 SAS 1.0 的 05 修订本相同,只有少量改进,信号速率不变。改进之处包括:

(1) 增加 4-wide 内部连接器(对应 SATA Ⅱ的 Ganged connector)。

(2) 简化的第一次突发(可选功能,不利用它对硬盘和磁带机等设备没有影响)。

(3) 链路/传输层重试:在 SAS 1.0 中,如果传输数据到设备的过程中发生错误,将导致整个传输终止,传输重试,这种机制与硬盘的行为相匹配,很少甚至根本不会影响整个系统的性能。然而,对流式磁带驱动器来说,重发数据却会因磁带回卷而造成性能显著下降。为了获得一个基本稳定的规范并加快 SAS 系统推向市场的速度,制订 SAS 1.0 时没有过多考虑这个问题。SAS 1.1 增加了链路/传输层重试,可以改善使用磁带驱动器时的性能。

(4) 改正了在 SAS 1.0 中发现的所有错误。

SAS 接口采用串行传送,采用点对点连接。一条 SAS 线缆内有两对数据线,分别用于发送和接收。线缆最大长度为 10 m。连接对象也不限于启动设备(主适配器)和目标设备(外围设备),还可以是扩展设备。整个系统可连接的设备达到 64 个,数据传输速率达到 3.0 Gb/s。

FC 接口 光纤通道(FC)是第一种能够实现数据块级的存储网络应用的网络体系架构。而光纤通道接口标准是由美国国家信息技术标准委员会(NCITS)的 T11 标准组织所开发的,定义了一个在网络基础设施上传输数据块级存储数据的多层体系结构。FC 还提供在网络中以千兆字节级的速率有效地传输数据块级存储数据的机制。FC 即用于高端存储设备,也可用于存储网络互联。

InfiniBand(IB)接口 InfiniBand 由 Intel、IBM、Cisco、Mellanox、Qlogic、SUN、Voltaire 等联合提出。1999 年建立的 IBTA(InfiniBand Trade Association)负责维护与制定。InfiniBand 是一种新型的总线结构,将服务器、网络设备和存储设备连接在一起的交换结构的 I/O 技术。InfiniBand 协议的主要特点是高带宽(现有产品的带宽 4xDDR 20Gbps,12xDDR 60Gbps, 4xSDR 10Gbps, 12xSDR 30Gbps、4xQDR 40Gbps,12xQDR 120Gbps)、低时延(交换机延时 140 ns、应用程序延时 3 μs、一年后的新的网卡技术将使应用程序延时降低到 1 μs 水平)、系统扩展性好(可轻松实现完全无拥塞的数万端设备的 InfiniBand 网络)。另外 InfiniBand 标准支持 RDMA(remote direct memory access),使得在使用 InfiniBand 构筑服务器、存储器网络时比万兆以太网以及 fibre

channel 具有更高的性能、效率和灵活性。InfiniBand 可以整合 fibre channel SAN、NAS 以及 iSCSI 进入服务器,目前主要用作存储系统互连和高端存储设备接口。

USB(universal serial bus)接口　通用串行总线,把计算机不同的外设接口统一起来,使用一个 4 针插头作为标准插头,通过这个标准插头,采用菊花链形式可以把所有的外设连接起来,并且不会损失带宽。USB 支持即插即用的特性。USB 的出现较好地解决了普通串口传输速度偏慢,连接设备有限的弊病。目前主要用于移动、便携式存储设备。

第一版 USB 1.0 是在 1996 年出现的,速度只有 1.5 Mb/s;两年后升级为 USB 1.1,速度提升到 12 Mb/s;2000 年 4 月,目前广泛使用的 USB 2.0 推出,速度达到 480 Mb/s;2008 年,USB 3.0 形成,最大传输带宽高达 5.0 Gb/s,也就是 640 MB/s。

iSCSI 接口　将 SCSI 接口技术与以太网技术结合,使存储设备中的数据可直接在以太网上传输的接口规范。

存储网络是个大的复杂的而且昂贵的网络。使用 SCSI,成本虽低,但传输数据时会受距离及路径的限制。若使用光纤接口,传输速率高,但结构复杂,投资大。用户需要结构简单、成本低、能够充分利用已有以太网环境完成数据存储工作的技术。因此,iSCSI 技术应运而生。2003 年 2 月 IETF 通过了 iSCSI 协议,并发布了 RFC 草案。

iSCSI 可以实现在 IP 网络上运行 SCSI 协议,支持在系统之间传送标准的 SCSI 命令。在系统之间的连接是通过标准的以太网的 IP 网络基础设施实现的,而不是通过 SCSI 线缆或光纤通道。

iSCSI 定义了可靠传输和使用 IP 路由的 TCP 中的 SCSI 信息包的封装。iSCSI 协议使得现有的 IP 网络(LAN、WAN 或 Internet)能传送整块的数据,而不用修改网络基础设施、主机软件或操作系统,也不用修改目标存储设备。

iSCSI 接口在存储区域网(SAN)上已得到应用。

2. 软磁盘驱动器接口　软磁盘控制器通常制成一块集成电路,直接装在主机板上。主机上的插座通过一条 34 芯扁平电缆与主机箱内部的驱动器相连,可连接两台驱动器。驱动器的插座用 34 针双列插座,其中奇数全接地。数据是串行传送,只占一位,其余是命令和状态信号。软磁盘接口的数据传输速率很低,有 250 kb/s 及 500 kb/s 两种(参见**软磁盘驱动器**)。

3. 光碟驱动器接口　常用的有 SCSI 接口、ATAPI 接口、EIDE 接口、USB 接口、PCMCIA 接口等。

ATAPI 是 ATA 的扩充。数据在 ATAPI 定义的“包命令”控制下,通过数据寄存器发送。数据传送可以通过 DMA 方式或 PIO 方式进行。

4. 磁带驱动器接口　常用的有 SCSI 接口、ATAPI 接口、EIDE 接口、USB 接口、PCMCIA 接口等。

(林兼　王芳)

waiwei kongzhiqi xinpian

外围控制器芯片(peripheral controller chip)

通过系统控制芯片与**中央处理器**(CPU)联系、用于处理外围总线信号的桥接芯片。在表示微机主板原理的框图中,CPU 芯片一般被置于图的最上部,外围控制芯片总是处于**系统控制器芯片**的下方。沿袭地图绘制中上北下南的习惯,应用中常将系统控制芯片称为北桥,而将位于北桥下方的外围控制芯片称为南桥(south-bridge)。如图 1 所示。

南桥是负责处理北桥与各种外围接口间通信的芯片。它既是 CPU 经由北桥访问各种外围接口的通道,同时又是各种外围接口经由北桥访问内存的通道。此外,南桥还负责处理各种外围接口产生的中断并负责将中断和对应的中断向量传送给北桥和 CPU。南桥中常见的外围接口控制器有:外围部件互连(PCI)控制器、PCIE 控制器、本地过程调用(LPC)控制器、直接存储器存取(DMA)控制器、磁盘(PATA/SATA)控制器、通用串行总线(USB)控制器、网络控制器、音频控制器、键盘控制器、实时时钟控制器、电源管理控制器、串行外围接口(SPI)控制器、SMBus 控制器等。由于南桥包含的外围接口大多与 I/O 相关,南桥又被称为 I/O 控制中心(Input/output controller hub)。一款南桥芯片通常可以和多款北桥芯片搭配使用。

随着半导体工艺的进步,单位面积可集成的晶体管数目日益增加,越来越多的功能正在被集成进南桥之中。在北桥功能逐渐转移到 CPU 内部的趋势下,PC 系统将逐步以 CPU + 南桥的形式出现。

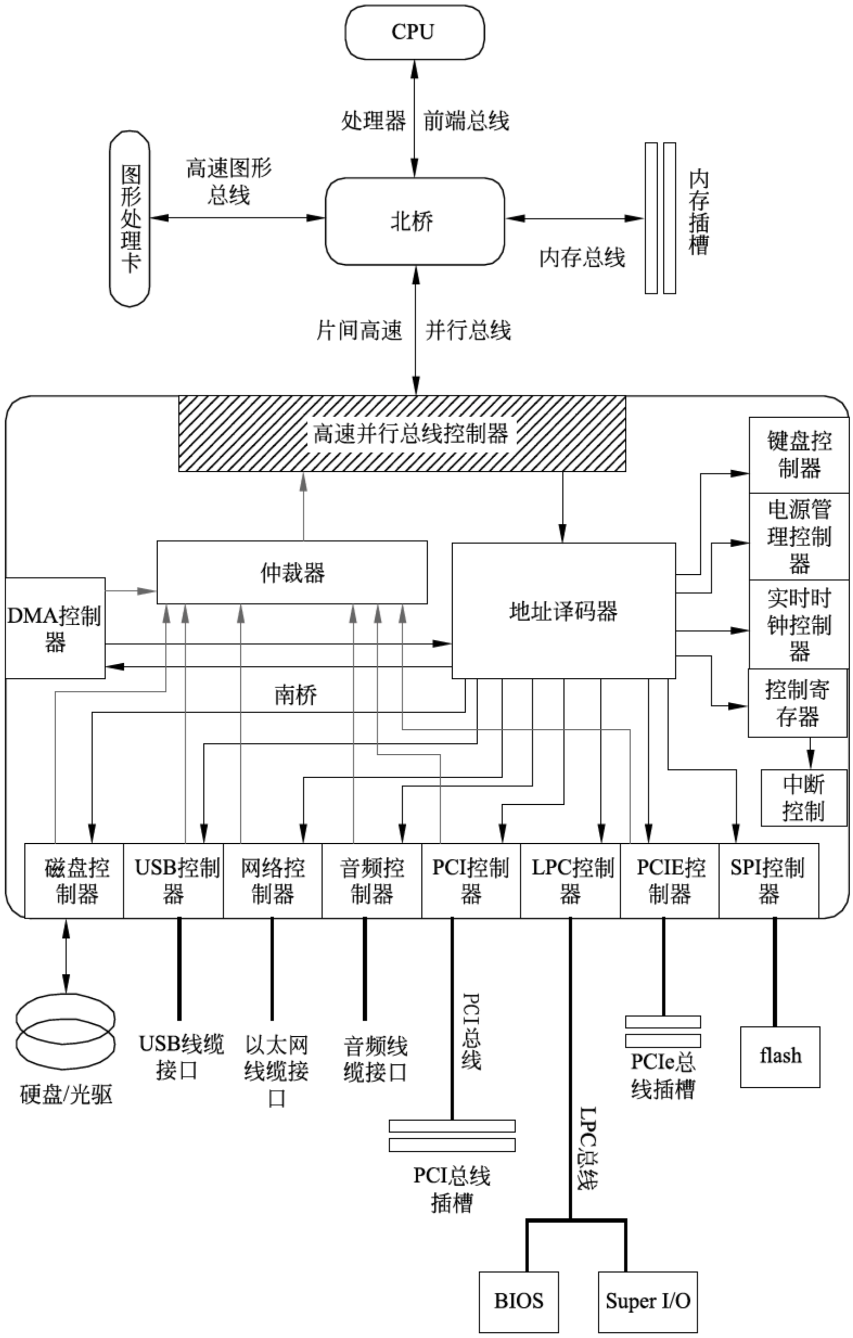


图 1 典型的南桥结构框图

参考文献

1. 陈国先. 计算机维护与维修. 3 版. 北京: 机械工业出版社,2001

2. 杨全胜,等. 现代微机原理与接口技术. 2 版. 北京: 电子工业出版社,2010 (蔡飞)

wanquan pianxu

完全偏序 (**complete partial order,CPO**) 具有某种完备性质的偏序集。它在 **λ-演算**和**指称语义**方面有着广泛的应用。

设 D 为集合, $\leqslant$ 为 D 上的二元关系。称 $\langle D, \leqslant\rangle$ 为偏序集,如果它满足: ① 对每个 $x \in D$ 皆有 $x \leqslant x$(自反性); ② 如果 $x \leqslant y$ 且 $y \leqslant x$,则 $x = y$(反对称性); ③ 如果 $x \leqslant y$ 且 $y \leqslant z$,则 $x \leqslant z$(传递性)。为方便起见,常把偏序集 $\langle D, \leqslant\rangle$ 简写为 D。如果偏序集 D 有最小(或大)元,则用 $\bot$(或 $\top$)表示,并称为 D

的底(或顶)。

具有最小元⊥的偏序集 D 称为平坦的,是指对任意的 $x,y\in D$ 皆有 $x\leq y$ 当且仅当 $x=\perp$ 或 $x=y$。

设 S 为偏序集 D 的一个非空子集。D 中的元素 x 称为 S 的上界,如果对每个 $s\in S$ 皆有 $s\leq x$。如果对 S 的每个上界 y 都有 $x\leq y$,则称 x 为 S 的上确界。如果 S 的上确界存在,则它必是唯一的。

设 S 为偏序集 D 的非空子集。如果对任意的 $x,y\in S$,都必有 $z\in S$ 使 $x\leq z$ 且 $y\leq z$,则称 S 为定向的。如果偏序集 D 满足:① D 有最小元;② D 的每个定向子集都在 D 中有上确界,则称 D 为一个完全偏序。

平坦偏序集总是完全偏序。其他一些完全偏序的例子有:具有最小元的有限偏序集;ω 序数连同定义在其上的小于等于关系"≤";集合 S 的幂集连同定义在其上的包含关系"⊆";实单位闭区间[0,1]连同定义在其上的小于等于关系"≤"。

设 $\langle D_1,\leq_1\rangle$ 和 $\langle D_2,\leq_2\rangle$ 都是完全偏序。若在 $D_1\times D_2$ 上定义二元关系"≤"如下:若 $x_1,x_2\in D_1$ 且 $y_1,y_2\in D_2$,则

$\langle x_1,y_1\rangle\leq\langle x_2,y_2\rangle$ 当且仅当 $x_1\leq_1 y_1$ 且 $x_2\leq_2 y_2$,

则 $\langle D_1\times D_2,\leq\rangle$ 是一个完全偏序,称为 $\langle D_1,\leq_1\rangle$ 与 $\langle D_2,\leq_2\rangle$ 的积。

设 $\langle D,\leq\rangle$ 为一个完全偏序。如果 $O\subseteq D$ 满足:

(1) 若 $x\in O,y\in D$ 且 $x\leq y$,则 $y\in O$;

(2) 若 X 为 D 的定向子集且 $\cup X\in O$,则 $X\cap O\neq\varnothing$;

则称 O 为 D 的 Scott 开集。若令

$$\mathcal{O}_D=\{O\mid O\subseteq D \text{ 且 } O \text{ 为 } D \text{ 的 Scott 开集}\}$$

则 $\langle D,\mathcal{O}_D\rangle$ 就构成一个拓扑空间,并称 $\mathcal{O}_D$ 为 D 上的 Scott 拓扑。这个拓扑空间是 T_0 空间,但不一定是 T_1 空间。

(王兵山　王水汀)

wanweiwang

万维网(world wide web, Web, WWW)　互联网上重要的信息检索工具,又称 Web。万维网通过超链接(Hyperlink)为用户提供对互联网中海量信息资源的便利访问。超链接分为**超文本**(hypertext)链接和**超媒体**(hypermedia)链接,简称链接。超文本链接允许用户通过选择文本中的关键字来获取进一步的相关信息,超媒体链接则指向图片、声音、动画以及电影等多媒体信息。万维网在互联网中基于客户端/服务器结构运行。服务器中的程序根据用户请求将各类相关信息存储并转发到另一台计算机上,而客户端上的程序为用户从服务器中获取信息。浏览器是最通用的万维网客户端,为用户从服务器中获取并显示信息。

万维网由许多互相链接的超文本文件组成。超文本文件又称为**网页**,由文本、格式描述和超链接构成,用超文本标记语言 HTML(HyperText Markup Language)书写。每一个超文本文件都有一个统一资源定位地址 URL(uniform resource locator),用于标识文件在互联网中的位置。每个超文本文件内部也包含多个由 URL 组成的超链接,指出其他超文本文件的位置以及浏览器如何处理它们。这样,用户就可以通过浏览器方便地访问万维网中的各类信息资源。

万维网服务器中向浏览器提供文档的程序称为 Web 服务器,多个内容相关的网页、动画、声音、视频、程序等文件组成网站(Web site),一个网站的起始网页称为首页(home page)。网站文件主要存储在 Web 服务器上,也可以存储在其他与互联网相连接的计算机上。

万维网由欧洲核子研究中心(CERN)的蒂姆·伯纳斯-李(Tim Berners-Lee)和他的同事于 1989 年共同发明。他们创建的超文本传输协议(HyperText Transfer Protocol, HTTP)制定了服务器和客户端之间的通信标准。他们于 1992 年 1 月发布了基于文本的 Web 浏览器。1993 年 2 月,美国的马克·安德森(Marc Andreessen)和伊利诺伊大学国家超级计算机应用中心(NCSA)共同研制并发布了第一个图形化的浏览器 Mosaic。Mosaic 具有类似于个人计算机上的图形界面,通过鼠标单击方式进行操作,使得万维网迅速为大众所接受。1994 年 4 月,安德森与合伙人共同创办了网景公司(Netscape Communications Corporation),他于 1994 年 12 月发布的网景导航者浏览器(Netscape Navigator)迅速在浏览器领域占据统治地位。至 20 世纪 90 年代中期,万维网已经拥有数百万活跃用户。

随着万维网的发展,2004 年出现了 Web 2.0 的概念,2006 年出现了 Web 3.0 的概念,这些概念没有严格而准确的定义。通常人们把单向信息发布的网站称为 Web 1.0,网站内容主要由制作者提供,用户以浏览阅读为主,典型代表包括各大门户网站、搜索引擎、大英百科全书在线(Britannica Online)等。Web 2.0 更注重用户的交互作用,用户既是网站内容的消费者、浏览者,也是网站内容的制造者,典型应用包括博客(BLOG)、RSS、百科全书(WIKI)、网

摘、社会网络(SNS)、P2P、即时信息(IM)等。Web 3.0基于Web 1.0和Web 2.0演变而来,也被称为**下一代万维网**,其定义和特征目前尚无定论,将更注重用户体验和个性化服务,具有语意理解和智能搜索等特征。

今天,万维网已经成为人类历史上最深远、最广泛的信息传播媒介;万维网技术把全球范围内的信息有机地组织在一起,使得全世界的人们以史无前例的巨大规模相互交流,推动了互联网的快速发展。

参考文献

1. Tanenbaum AS. 计算机网络. 4版. 北京:清华大学出版社, 2004

2. Encyclopedia Britannica online, 2014

(胡道元 张蓓)

wanweiwang shuju guanli

万维网数据管理(Web data management)

对万维网中的各种复杂数据和信息进行有效地组织和集成以方便而准确地查询和发布的一种数据管理技术。它是融合了网络技术、数据库技术、信息检索技术、多媒体技术和数据挖掘技术等多个领域技术的极具挑战性的一个新兴的研究领域。

除千变万化的网页本身的内容(包括文本或图形数据)外,万维网数据还包括网页的内部结构、网页之间的链接结构、如何使用网页的描述数据、用户简档(统计信息和注册信息)以及从cookie中获取的信息等。万维网的一个关键特征是它为存储的文档提供了超文本结构,文档包含链接,链接指向其他也存储在万维网上的文档和资源。万维网的开放性和用户的随意性使得信息资源的结构很复杂,大量数据是半结构化和无结构化的。万维网数据源具有不同的数据类型(数据异构)、不同的模式结构(模式异构)和不同的语义内涵(语义异构),它还具有分布分散、动态变化和规模巨大等特点。这些都给万维网数据管理的研究和应用带来了挑战和机遇。

万维网数据管理主要包括万维网信息查询、万维网数据集成、万维网信息发布和万维网数据挖掘等。

万维网信息查询(Web information query) 指根据用户的请求在有效的数据组织模式下从万维网中找出准确信息的机制和过程。目前主要使用基于搜索引擎的关键词索引技术来进行信息查询。

万维网数据集成(Web data integration) 提供用户对多种异构数据源透明、一致和实时的访问,尽量将万维网上的诸多数据源中的信息构成为一个为用户可用的整体。透明性是屏蔽底层数据源的差异,让用户感觉数据来自一个数据源;一致性是消除数据源之间存在的结构异构和语义异构;实时性则指访问到的数据是最近更新过的并能较快的响应查询请求。目前已涌现了很多数据集成技术和数据集成中间件,具有代表性的技术是ETL技术和I3解决方案。

万维网信息发布(Web information publishing) 把信息发布者需要发布的信息进行编辑后发布到万维网中以及把万维网上的信息按用户的需求自动发送给目标用户。

万维网数据挖掘(Web data mining) 使用数据挖掘技术发现万维网上的信息结构和模式,包括万维网内容挖掘、万维网结构挖掘和万维网使用挖掘三类。万维网内容挖掘应用数据挖掘技术分析网页,发现万维网上的有益信息和模式,是对基本搜索引擎的扩展;万维网结构挖掘是从网页的实际组织结构中获取信息,为万维网(或者其中的一部分)组织建立一个模型,使用该模型来对网页分类或者为网页建立相似性度量;万维网使用挖掘根据网站的访问日志,识别用户的访问模式,把用户按相似的访问进行分组。

由于万维网上的信息五花八门,没有统一的表示,因此给数据管理带来了困难。如果网络上的资源在创建之初就使用标准的元数据来描述,就可以省去很多麻烦。**资源描述框架**(resource discription framework,RDF)给出了有效的元数据解决方案。它是W3C提出的用于描述万维网资源的标准,可以用它来描述和注解万维网中的资源,并且向计算机系统提供理解和交换数据的手段。

万维网上的数据越来越多地用可扩展标记语言XML描述,XML数据管理的研究也正在万维网数据管理中显示出它的优势。

数据库技术、数据仓库技术、数据挖掘技术、XML数据管理技术以及新一代分布式数据库技术等都致力于万维网数据管理和应用的研究,试图最大限度地实现对分布在互联网中的各种不同数据源的透明访问和信息抽取,让万维网中的数据和信息更好地为应用服务。

参考文献

Bhowmick S S, Madria S K, Ng W K. Web data management: a warehouse approach. Springer-Verlag, 2004

(周龙骧 王翰虎 孟小峰)

wanzhaowei yitaiwang

万兆位以太网(10 gigabit Ethernet)　万兆以太网规范包含在 IEEE 802.3 标准的补充标准 IEEE 802.3ae 中,它扩展了 IEEE 802.3 协议和 MAC 规范,使其支持 10Gb/s 的传输速率。除此之外,通过 WAN 界面子层(WAN interface sublayer, WIS),10 千兆位以太网也能被调整为较低的传输速率,如 9.584 640 Gb/s(OC-192),这就允许 10 千兆位以太网设备与同步光纤网络(SONET) STS-192c 传输格式相兼容。

(1) 局域网相关标准:10GBASE-SR,短距,多模光纤,850 nm 激光器,传输距离在几十至几百米之间。10GBASE-LR,长距,单模光纤,1310 nm 激光器,传输距离在 20 km 以内。10GBASE-ER,扩展长距,单模光纤,1550 nm 激光器,传输距离可达 40 km。

以上三种标准在物理编码子层(PCS)采用了 64B/66B 格式,因此速率是 10.3125 Gb/s。10GBASE-LX4),结合了 8B/10B 编码和粗波分复用技术,4 个分离的激光器,每通道的速率 3.125 Gb/s,在单模光纤上最长传输距离可达 10 km。

(2) 广域网相关标准:10GBASE-W,万兆以太网广域标准,定义了与 SONET OC-192c 和 SDH VC-4-64c 兼容的帧格式,有效速率 9.95 Gb/s。

(3) 铜缆互连技术标准:10GBASE-CX4,采用与 Infiniband 类似的铜缆互连技术,每通道速率 3.125 Gb/s,工作距离可达 15 m。802.3an-2006,10GBASE-T,采用双绞线铜缆互连,距离可达 100 m。

参考文献

敖志刚. 万兆位以太网及其实用技术. 北京:电子工业出版社, 2007　　(全耀辉)

wangge jisuan

网格计算(grid computing)　通过**互连网络**,将不同空间位置、不同类型的物理与逻辑资源以开放和标准的方式组织起来,通过资源共享和动态协调,来解决不同领域的复杂问题的分布式和并行计算。网格计算改变了人们对计算的传统看法,也改变了人们传统的解决问题的方式。网格计算是借鉴传统电力网的概念提出来的。在网格计算中,人们能使用一种来自网络的计算能力与资源,无须知道网格资源的确切提供者以及提供者的地理位置,只需集中精力考虑如何使用这些资源与能力来解决更富有挑战性的问题。

发展简史

网格计算的研究可以划分为三个阶段:

第一个阶段是从 20 世纪 80 年代末到 90 年代初期,是网格计算的萌芽阶段。这时出现了比较成熟的千兆网技术、元计算系统和**集群计算**系统,这些都为网格计算的出现提供了必要的条件。

第二个阶段是从 20 世纪 90 年代中到 90 年代末,是网格计算的早期研究和实验阶段。此时出现了一些实验项目,比如 I-WAY 等,还有不少学术性的研究项目和应用,这些项目对以后的网格计算研究有很大的影响,比如 Globus。

第三个阶段是从 21 世纪初到现在,是网格计算迅速发展和广泛应用的阶段。这时网格计算的概念已经被广为接受,越来越多的国家、地区与组织都在积极开展各自的网格计算项目,一些大型的世界性网格组织如 GGF 和重大网格计算应用已经开始发挥作用。但是还存在不少问题需要解决,突出的问题就是标准化的问题,当大量的网格计算系统和应用出现以后,如何制定一个统一的标准来规范化这一新兴领域是一个亟须解决的问题。开放式网格服务体系结构(OGSA)就是这种努力的结果,有不少组织在开展这方面的工作。

组成与特点

网格系统一般从下到上分为三个主要组成部分(如图 1 所示):①网格基础构件;②网格管理与服务系统;③网格应用。

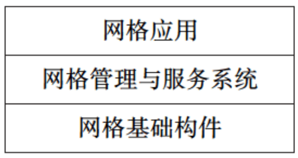


图 1　网格系统层次结构

网格基础构件包括网络和其他的各种网格物理资源结点两大组成部分,网格物理资源由各种不同类型的计算资源、数据资源、信息资源、仪器设备等组成。这些资源所处的地理位置是分布的。网格资源结点之间通过互连网络或其他高速网络连接起来,组成网格向外提供各种资源与服务,实现资源的共享与协作。同时每个物理资源结点又是一个具有自主功能的相对独立的系统,可以有自己独特的管理和使用策略。网格管理与服务系统通过实现对网

格基础构件的统一管理,为各种不同的网格应用提供服务。网格管理包括安全性保证,资源的动态发现、管理与协作,跨地域的任务调度,异构系统的协作,网格协议的管理与实现等。网格服务是各种网格功能的综合体现,各种不同的应用是通过具体的网格服务来使用网格功能的。网格应用是网格计算的最终目的,通过网格应用可以解决复杂的更具有挑战性的问题。

网格计算一般具有如下特点:①灵活性与扩充性,即网格计算的方式不仅十分灵活,而且网格资源是可以不断扩充的,网格计算系统是可扩展的;②局部自治性,网格资源结点是一个相对独立的系统,具有独立自主性;③全局名字空间,即在不同的地方,用户都可以用相同的名字和密码对网格计算系统进行访问,不会因为地理位置和具体系统的不同而有所改变;④透明访问,即网格资源的位置、类型等特性对用户是透明的;⑤高性能与高质量,即网格系统必须能够提供更高的性能,同时对提供的服务保证高质量;⑥安全性,即对网格系统的使用是有安全保障的,不用担心秘密或者私人信息被他人截获或者泄漏;⑦异构性与可移植性,即网格系统可以容纳不同类型的资源,同时保证网格应用可以方便地在不同的网格系统上移植;⑧容错性,即网格系统必须能够对用户或者系统的异常错误能够容忍和恢复,不致造成巨大的损失;⑨动态性,即网格系统应该能够动态适应底层计算资源和上层网格应用的变化。

网格概念的核心与实质

网格概念的核心是资源(包括服务)以及对资源的使用,这里的资源包括计算机、数据库、仪器设备、信息服务等极其广泛的内容。网格概念背后的实质,是在打破传统的强加在资源之上的种种限制的基础上,为使用者提供一种前所未有的高级服务。所谓打破对资源的限制,包括下面几个方面的含义:①资源的网格化,即将资源从特定的地理位置的束缚中解放出来,使得该资源可以通过网格输送到任何角落,达到网格资源与地理位置分离的目的;②网格资源的协调,即在一定的规则约束和管理下,任何网格资源都可以实现相互协作,打破不同资源之间在广泛共享与协作方面的障碍;③网格资源的融合,可打破原来在资源能力和资源类型方面的限制。因此,网格系统提供的资源是增强和放大后的可以动态组合与使用的资源。

关键问题与技术

网格计算的关键问题包括网格体系结构与网格标准化问题。网格作为一个庞大的信息基础设施,必须为它设计一个坚实可靠的体系结构,网格体系结构是支撑网格有效正常运转的基石,因此,必须重点解决网格体系结构所面临的种种问题。

网格计算面临的另外一个问题就是标准化的问题。网格计算走向实用,必须建立一个共同遵守的有效的标准。只有这样,才能够完成已有系统的网格化改造,才能够将新开发出来的系统方便地融入到已有的网格系统之中,避免由于标准不统一造成在效率、性能、共享等方面的问题。

网格计算还存在着一些关键的技术问题有待解决。包括:网格资源的监测与发现技术,随着网格系统包括的资源越来越多,网格资源的准确发现、定位与监测是正确网格调度的前提,是网格资源得以有效使用的前提;网格资源管理,通过对网格资源的动态高效的分配、调度以及管理,可以实现对网格资源的有效、充分使用,并为网格应用提供高质量的支持;传统资源的网格化包装问题,各种传统的物理资源,只有通过特定的形式转化为网格可以使用的资源,才能够充分发挥其作用;各种网格应用门户的开发技术,用户是通过网格门户来使用网格的,因此,网格门户的开发与实现技术直接影响到网格能否被广泛接受和使用。

研究类别

关于网格计算的研究,可以从如下几个方面进行划分。

第一个方面是关于网格基础设施的研究和开发项目,其中最有代表性和影响力的就是 Globus 项目。该技术目前支持的几种典型应用包括:①分布式超级计算;②快捷仪器,通过访问数据文档和提供在线处理能力而增强科学仪器的功能,比如 X 射线 CMT 应用;③桌面超级计算,即通过桌面系统就可以访问到超级计算能力;④远程沉浸,通过将模拟技术、虚拟现实技术和协同工作环境相结合,提供一个共享的虚拟设计空间,这种应用对计算和网络的性能要求很高。

第二个方面是网格系统建设的研究,包括简化用户与网格资源之间的交互,分布的科学团体之间的支持密集计算和大数据量分析的计算基础设施,实现分布式的高性能计算和对科学与工程计算中大粒度数据的管理等研究。

第三个方面是网格计算应用和相关库开发方面

的工作,由于网格的最终目的还是为了支持各种应用,因此,面向不同的应用领域开发网格应用和相关的库也是十分重要的工作。

第四个方面是商用网格计算的研究,已经有了一些初步的产品。

几个典型的网格计算试验床

大型的网格计算试验床可以支持大量的用户群,而且能够使得不同学科的科学与工程研究人员在试验床上开展富有成效的研究工作。主要有以下几个研究项目:

(1) 美国国家技术网格　它由 NPACI 和 NCSA 两大组织负责建立该网格计算系统。他们运用 Globus 将各种设施和资源,包括超级计算中心,重点研究实验室,学院和大学的校园连接起来,建立网格计算系统,并且鼓励使用原型网格计算系统,支持分布式的科学与工程计算与应用,形成全美范围的计算基础设施的基础,就像当初的 NSFnet 对后来的互连网络所起的作用一样。

(2) 欧洲的数据网格　它由 6 个主要的联合伙伴(CERN,CNRS,ESRIN,INFN,NIKHEF,PPARC)和 15 个相关的成员组成,它在网格技术的支持下希望达到如下目的:①建立一个研究性网络,开发相应的技术,以建立一个更大规模的全球数据网格;②让真正的用户介入,通过大规模的应用来验证这种技术的有效性;③展示建造、连接和有效管理大规模通用目的的低成本数据密集型集群式计算机的能力。

(3) NASA 的 IPG　运用 Globus 工具,将加入到其中的组织的超级计算机和存储设备连接起来形成一个单一、无缝的计算环境。在政府、学术界和工业界的共同努力下,IPG 可以帮助美国科学家协同解决 21 世纪面临的重要问题,正如 Web 使得任何地方的信息都可以从所有的地方访问一样,IPG 希望给美国的研究者和工程师,在他们需要的任何时候,提供远距离超级计算资源和数据仓库的能力。

(4) ASCI 的分布资源管理(DRM)试验床　美国能源部的 Lawrence Livermore、Los Alamos 以及 Sandia 三个重点实验室在从事 ASCI 的研究,它是美国核计划的重要组成部分,这些实验室在建造 DRM 的试验床,用于管理所有实验室的计算资源。使用的是 Globus 工具系统,支持 Kerberos 安全。

(5) GUSTO 试验床　用来测试 Globus 工具箱。它由一些联合伙伴共同建造,到 2000 年 2 月为止,它已包括了 23 个国家的 125 个地点,是最大的计算环境之一。

发展趋势

网格计算正处在急剧膨胀和发展的阶段。虽然网格计算的技术难点还没有完全解决,但是已经跨越了以前的探索与实验性研究的阶段。其研究中心将转移到网格计算标准的制定和开辟真正有效的网格计算应用。

OGSA 代表了网格计算体系结构的新发展和标准,它将网格中的一切资源都抽象为服务,借鉴了 Web 服务的框架,并结合已有网格研究的成果,将传统的科研领域的研究和商业领域的已有成果有机集成,形成一种同时适合两者的体系结构。

网格计算的研究出现了多样化的趋势,从原来的主要以计算为主的研究发展到全方位的网格服务研究。网格系统的应用领域一方面在扩展,另一方面又在细化,因此出现了不少专用的网格系统,如地震网格,商用网格等。

参考文献

1. Foster I, Kesselman C. The Grid 2: blueprint for a new computing infrastructure. San Francisco: Morgan Kaufmann Publishers,2003

2. Foster I, Kesselman C, Tuecker S. The anatomy of the grid: enabling scalable virtual organizations. International Journal of High Performance Computing Applications, 2001, 15(3): 200-222

3. Foster I, Kesselman C, Nick J M, et al. Grid services for distributed system integration. IEEE Computer, 2002,35(6): 37-46

(都志辉)

wangge qumian

网格曲面(mesh surface)　用来表示或逼近空间曲面形状的一组多边形集合。也叫作非结构网格,是曲面离散表示的重要手段之一,尤其是在**计算机图形学**和实体造型方面有着广泛的应用。为了便于绘制、编辑和传输,多边形通常由三角形、四边形或者简单的凸多边形组成,但是也可以扩展到更一般的凹多边形。

表示网格曲面的数据结构有很多种。面片-顶点表示方法使用顶点坐标的列表和多边形面片的顶点序号。翼边表示方法在每条边上记录与其直接相连的两个顶点、两个面片以及其他四条边。半边表示法将翼边结构中每条边一侧的信息单独存储。顶点-顶点表示方法对每个顶点仅记录其相邻的其他顶点。此外还有其他一些网格曲面数据结构。每种

结构各有其优、缺点，通常根据实际需要来选择。在需要快速获取拓扑信息（如边或者邻接面等）时，较为复杂的翼边或半边结构比较好；在面向硬件绘制时，则常常采用简单紧凑的结构如顶点表示。

网格曲面可以通过**点云**重建、多边形建模等途径获得。对于已有的网格曲面，常见的操作包括修补、光顺、简化、形变、细分、参数化、重网格化等，这些操作在计算机图形学和实体造型中有着重要的意义。

网格曲面是空间曲面的离散分片逼近形式，因此网格曲面几何属性（如法向、曲率等）的计算方法可由对这些空间曲面几何属性进行离散逼近获得。尽管这一思想是显而易见的，但由于大多数时候网格曲面所对应的连续空间曲面是未知的，这给具体的离散化计算方法带来了一定的困难。

参考文献

1. Botsch M, Pauly M, Kobbelt L, et al. Geometric modeling based on polygonal meshes. ACM SIGGRAPH Course Notes, 2007

2. Botsch M, Kobbelt L, Pauly M, Alliez P, Lévy B. Polygon mesh processing. Natick, MA: A K Peters, 2010

（黄劲）

wangge qumian canshuhua

网格曲面参数化（parameterization of mesh surface） 构造从三维曲面一部分到二维欧氏空间区域映射的过程和技术。它是几何处理过程的重要一环。网格曲面参数化对如何用一张曲面逼近另一张机械制造中特定曲面的数值仿真、曲面的重采样和编辑、曲面的**纹理映射**等都有非常重要的意义。

根据参数域的不同，参数化通常可以分成平面参数化、基网格参数化和球面参数化等几种。

平面参数化是最简单和常见的形式。多数方法通过某种能量函数的极小定义来构造出具体的映射函数。这种能量可以是度量保持长度，降低变形扭曲，保持角度或者保持面积，也可以是弹性能量。为使得所定义的能量极小，通常可以通过确定边界和交互约束，将约束优化问题转化为求解以投影坐标为未知量的大型线性方程组。幸运的是，所得到的大型矩阵一般较稀疏，利于快速数值求解。也有研究者通过两步法，在映射函数之后再添加一个变形函数，以减少参数化的整体变形问题。同时，参数化可看成降维问题的一个特例。如有些研究者通过求解曲面上顶点间的测地距离，然后利用经典 MDS 将三维网格降维映射到二维平面，得到一种自由边界的参数化方法。另外，也有研究者直接采用构造映射的方法实现网格的参数化。同时，平面参数化还可以分成固定边界和可变边界两种。早期的研究主要集中在固定边界上，但是当预定义边界和实际需要相差较大时会导致比较严重的变形问题，所以近年来许多方法将边界也作为未知变量和参数化同时优化求解。另外，受限于拓扑结构，当输入网格的拓扑较为复杂时，往往无法对模型直接作参数化处理。合理的做法是使用各类分片技术，然后再对各子面片进行参数化。

基网格参数化需要首先构造一个和原网格拓扑一致的简化网格，然后将原网格顶点参数化到基网格的平面上。一些研究者还提出迭代松弛等后处理方法来调整参数化在边界上的光滑性。基网格参数化的优点在于保持了拓扑一致，因此可以适应高亏格的网格。同时由于在构造时较好地保持了原网格的几何信息，通常形变较小。

球面参数化主要针对亏格为 0 的封闭曲面。相比前两种参数化，相关研究起步较晚。一类常见的方法是首先将所有顶点都投影到网格模型的最小包围球上，然后不断在球面上松弛顶点，最终实现球面参数化。

参考文献

1. Aksoylu B, Khodakovsky A, Schröder P. Multilevel solvers for unstructured surface meshes. SIAM Journal on Scientific Computing, 2005, 26(4): 1146-1165

2. Zigelman G, Kimmel R, Kiryati N. Texture-mapping using surface flattening via multidimensional scaling. IEEE Trans on Visualization and Computer Graphics, 2002, 8(2): 198-207

3. Khodakovsky A, Litke N, Schröder P. Globally smooth parameterizations with low distortion. ACM Trans on Graphics, 2003, 22(3): 350-357

（张宏鑫　黄劲）

wangge qumian jianhua

网格曲面简化（simplification of mesh surface） 对给定网格构造相对粗略的逼近版本，使能很好地保持原有网格几何特性的技术和算法。它是网格处理的重要手段之一。由于激光扫描仪等获取的网格数据往往数据规模庞大，超过了现有计算系统的处理能力，为使得网格数据得到有效的存储、

传输和计算,网格简化成为重要研究内容。除了作为网络传输、存储和显示用途外,网格简化也是许多网格参数化方法的预处理步骤,同时还可用来实现模型的多层次(LOD)结构。由于三角形网格简单易用,是数字几何处理中最广泛采用的数据形式,也是三维网格模型简化的主要对象。

网格曲面简化通常按某种规则,逐步地合并和删除一类拓扑元素来达到数据简化的目的。按操作的中心对象可划分为基于顶点删除、边塌缩、面删除和顶点归一化等几类简化方法。基于顶点删除的简化方法,每次缩减一个顶点及其相关的边和面,然后使用 Delaunay 三角化方法来填补顶点删除后所形成的空洞。类似地,也可以以边为中心,实现简化。边塌缩算法则只需合并当前的边所连接的顶点和删除两个退化的三角形;半边塌缩算法只将边连接的一个顶点作为规则判别对象,而将另一顶点塌缩吸收。面删除算法中较典型的是三角形删除法和面合并算法。比较特殊的是顶点归一化方法,它并不要求待合并的两点有边相连。

简化过程中的误差控制规则非常重要,它可以用于选择更好的删除元素和简化效果,度量简化过程的质量,以及给出简化的终判条件。目前已有许多很好的简化方法,如基于二次误差度量的简化方法、以边塌缩和顶点分裂为互逆操作的渐进网格框架、通过延法线方向度量向内和向外的偏移量的包络法,以及各种保持体积或者表面积、曲率、特征边的个数和入度等度量的简化方法。另外,除了基于几何信息的简化外,许多研究者还考虑了基于表面纹理和颜色的简化方法。总之,简化过程的关键在于选取合适简化的单纯形,添加所需的新单纯形,并确定简化规则。

当网格数据规模超大时,部分研究者采用了分片载入处理的策略使得数据在现有计算框架之下得以处理。

参考文献

1. Garland M, Heckbert P. Surface simplification using quadric error metrics. In: SIGGRAPH 97 Conference Proceedings, Annual Conference Series, 209-216. ACM SIGGRAPH, Addison Wesley, August 1997

2. Paolo Cignoni, Claudio Rocchini, Claudio Montani, Roberto Scopigno. External Memory Management and Simplification of Huge Meshes. IEEE Trans on Visualization and Computer Graphics, 2003, 9(4): 525-537

(张宏鑫　黄劲)

wangguan

网关(gateway)　在采用不同体系结构或协议的网络之间进行互通时,用于提供翻译地址、转换协议、变换数据格式等功能的设施。又称网间连接器、协议转换器。

目前,网关主要分类如下:

(1) 协议网关　协议网关通常在使用不同协议的网络区域间做协议转换,如在使用不同帧类型或时钟频率的局域网间提供转换以达到互联,在 IPv4 和 IPv6 网络之间对分组报头的地址进行转换,同时根据协议的不同对分组做相应的语义翻译。

(2) 应用网关　应用网关是在使用不同数据格式的应用程序间做翻译数据,典型的应用网关接收一种格式的输入,将之翻译,然后以新的格式发送出去。

(3) 安全网关　是指设置在不同网络或网络安全域之间的一系列部件的组合的统称,它可通过监测、限制、更改跨越安全网关的数据流来实现网络和信息的安全。

(毕军)

wangji xieyi

网际协议(internet protocol, IP)　Internet 采用的一种无连接传送协议。在网络分层模型中所处的位置如图 1 所示。

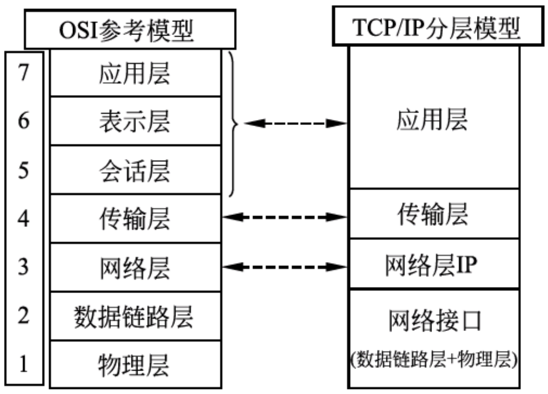


图 1　OSI 参考模型和 TCP/IP 分层模型比较

IP 协议屏蔽物理网络特性,提供统一的逻辑网络即 IP 网络,使异构的物理网络能够互相通信。IP 协议提供三种功能:

(1) IP 协议定义在互联网中数据传送的基本单元为 IP 数据报

互联网的基本传送单元是 IP 数据报,包括报头部分和数据区部分。目前最为广泛使用的报头的版本号为 4(IPv4 协议)。由于 IPv4 地址的耗尽问题,

报头版本号为 6(IPv6 协议)得到越来越广泛的应用。IPv4 报文头和 IPv6 报文头如图 2 所示。

IPv4 报文头和 IPv6 报文头中功能和名称相同的域有:版本号、IP 源地址和 IP 目标地址,但 IPv4 的地址为 32 位(二进制),IPv6 的地址为 128 位(二进制)。IPv6 报文头和 IPv4 报文头中具有类似的功能,但名称和位置均改变的域有:IPv4 服务类型变为 IPv6 流量分级;IPv4 的总长变为 IPv6 的载荷长度;IPv4 的存活时长变为 IPv6 的转发计数;IPv4 的协议类型变为 IPv6 下一个头;IPv4 的校验和在 IPv6 中被取消;IPv6 不允许路由器对数据报分片,因此 IPv4 中的标识、标志位和分片偏置被取消;IPv4 中的选项被 IPv6 下一个头取代。此外,IPv6 中增加了流标识。

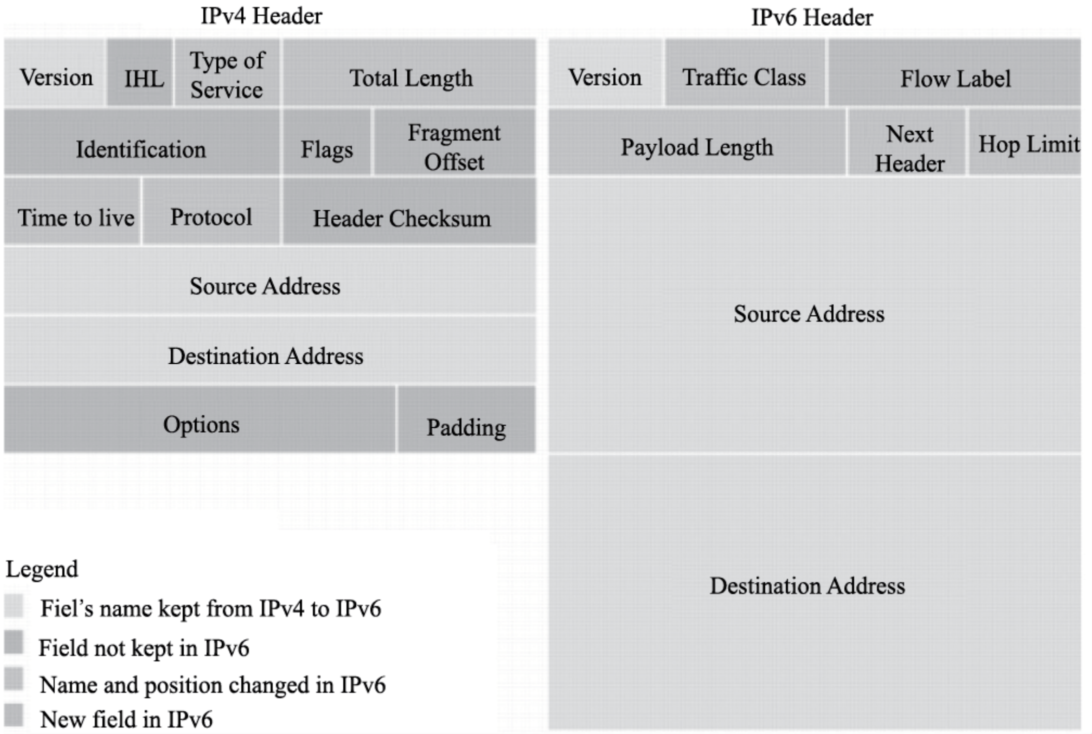


图 2 IPv4 报文头和 IPv6 报文头

数据报通过底层的物理网传输,为使互联网传送更有效,要保证每个数据报都用单独的物理帧传送。不同类型的物理网络对一个物理帧可传送的数据量规定了不同的上限,称为网络最大传送单元(MTU)。如数据报在传输过程中其数据报的长度大于某个传输路径上的 MTU,则根据数据报文头的相关参数,数据报可能被分片。在这种情况下,分片的数据报只能在到达目的主机后重组。

(2) IP 协议根据 IP 目标地址选择路由,即数据报传送的路径

在互联网中,路由选择是指选择一条路径来转发数据报文的过程。可以把路由选择分成两种方式,即直接传送和间接传送。由于 IP 地址可以通过前缀长度把 IP 地址分为网络标识和主机标识两部分,通过判别网络标识就能确定 IP 源地址和 IP 目标地址是否在同一网络上。

在同一网络上,两台机器之间 IP 数据报的传送不涉及路由器,为直接传送。发送方将数据报封装在物理帧中,将目标 IP 地址和目标的硬件地址绑定在一起,并将产生的物理帧直接传送到目标机器。

在不同网络上,数据报的传送采用间接传送方式。发送机器将数据报传送至路由器,一旦物理帧到达该路由器,封装的数据报就被提取出来,根据数据报头中的 IP 目标地址和路由表选择通往 IP 目标地址的下一台路由器。这个过程直到数据报到达某台可以将其直接传送到目标机器的路由器为止。为了进行路由选择,需要路由选择表,该表储存各个目标网络以及如何到达该目标网络的下一跳的 IP 地址。路由表可以静态配置,或通过在路由器之间运行路由协议自动生成。

(3) IP 协议包含了一组数据报处理、差错信息发生以及数据报丢弃的传送规则

通过路由器转发数据报，IP 协议提供无连接数据报传送服务。假如路由器不能正确选择路由或传送数据报，或者它检测到一个异常条件影响它转发数据报，路由器需要通知源站点采取措施避免或纠正出现的问题。为了使 TCP/IP 互联网中的路由器能报告差错或提供有关意外情况的信息，在 TCP/IP 协议中设计了一个特殊用途的报文机制，称 Internet 控制报文协议（在 IPv4 为 ICMP，在 IPv6 为 ICMPv6）。它是 IP 的一部分，并在每个 IP 实现中都是必需的。

参考文献

1. Jon Postel. RFC791: Internet Protocol. 1981

2. Deering S, Hinden R. RFC 2460: Internet Protocol, Version 6 (IPv6) Specification. 1998

3. Li X, Bao C, Baker F. RFC6145: IP/ICMP Translation Algorithm. 2011

（李星　包丛笑　胡道元）

wangluo anquan

网络安全（network security/network safety）　一门涉及计算机技术、网络技术、通信技术、应用密码学技术、管理技术等多领域的综合性学科，重点解决分布式计算环境（主要是互联网环境）中，网络设施与网络信息资源设计、实现和使用过程的安全性问题，以保障网络数据的传输安全、网络用户与系统资源和信息资源的访问安全以及网络应用系统与网络基础设施的运行安全。由于**计算机网络**是由计算机互联构成，因此网络安全领域的内容最初主要来源于面向单机保护的系统安全领域，包括计算机的外部环境或自身问题导致的威胁，包括物理破坏、电磁泄漏和软件故障等。多用户操作系统出现之后，人们开始关注使用的安全，如操作系统中的身份鉴别和访问控制；数据内容的保护，如数据的加密和完整性保护。网络安全所涉及的安全机制和安全服务在计算机网络环境中因不同的需求变化而产生新的发展，构成不同时期特定的网络安全基本内容。由于互联网已经日益渗透到全球政治、经济和文化生活的各个方面，各个不同局部网络的安全都有可能彼此相关，因此今天对网络安全的讨论默认是面向互联网的。

计算机网络技术的早期发展侧重于其开放性和灵活性，以促进信息交流与共享，网络用户的覆盖范围有限且彼此信任，因此在安全性方面考虑甚少。然而，随着互联网络覆盖面的扩大，网络安全问题开始显现。1986 年，时任美国劳伦斯国家实验室系统管理员的天文学家兼作家克利福德·斯托尔（Clifford Stoll）使用蜜罐技术成功地发现了一个窃取计算机信息的国际黑客，这是第一个公开报告的互联网安全事件，表明网络可以用于破坏性目的。1988 年底发生的莫里斯蠕虫事件的严重性引发了业界对计算机网络安全问题的高度重视。这个网络安全事件不仅开拓了攻击者的视野（可以利用安全漏洞实现自动的攻击和攻击传播），也开拓了防御者的视野（计算机应急响应组织应运而生）。

20 世纪 90 年代之后，互联网逐步商业化，网络应用涉及各行各业，用户覆盖社会各类人群，从而在网络资源的使用约束和网络基础设施运行保护方面产生强烈的需求，一些发达国家启动了大规模的网络安全领域的科学研究计划和安全产品评估的标准制定，推动了网络安全领域中各个单元技术的快速发展。恶意代码的传播渗透能力和攻击能力越来越强，种类繁多。对网络安全漏洞的研究与利用全面展开，网络攻击的系统性和可继承性日趋成熟。与此同时，也出现了包括**网络入侵检测**系统、**防火墙**、各种网络环境下的身份认证系统、公钥管理体系（PKI）、用于保护 Web 访问的 SSL 协议、安全漏洞扫描系统、防病毒软件等在内的一系列网络安全防御与安全管理技术和相应的产品。

进入 21 世纪之后，网络安全领域的研究日趋深入，发达国家纷纷将其纳入国家战略的高度予以规划和实施。在互联网环境中，网络攻击开始进入产业化阶段，一方面以追逐经济利益为目标，形成地下产业链；另一方面以服务于政治或国家利益为目标，构成有组织的行为。与此相适应地，网络安全防御与安全管理技术也在逐步完善，产品功能设计更为合理，并趋于一体化。例如个人计算机中的防病毒软件从单纯的计算机病毒检测进化为集个人防火墙、入侵检测、恶意代码防范等多种功能在内的综合性系统安全防御系统，并可在其厂家提供的中央服务器的支持下构成分布式的覆盖更大范围的网络安全监测与防御体系。从系统安全和信息安全领域借鉴各种相关技术而逐步发展起来的网络安全领域正在形成自己的理论体系和技术框架（信息空间安全，Cybersecurity）。

网络安全的目标是要能够应对计算机网络所面临的威胁，保障网络的真实性、可用性、完整性、保密性和拥有性。网络的真实性是指网络中实体（硬件、软件、数据、用户等）的标识是真实且可验证的。

网络的可用性是指即使存在干扰,网络也可按预期的能力工作,完成指派的任务,获得正确的结果。网络的完整性是指数据未经授权不能进行改变的能力,即信息在存储或传输过程中保持不被修改、不被破坏和丢失。网络的保密性是指信息不泄露给非授权实体的能力。网络的拥有性是指网络管理者对网络资源有完整的控制能力,不被网络攻击者劫持或实际占有。

网络安全领域的内容大致可划分为与网络攻击相关、与网络防御相关以及与网络安全管理相关。

计算机网络所面临的安全威胁是由网络攻击造成的,而网络攻击所依据的是网络及其应用在设计、实现和使用过程中出现的安全漏洞(又称安全缺陷)。各种类型的网络黑客是网络攻击的发起者,但不一定是网络攻击的实际执行者。网络黑客可以借助通用的软件工具或系统功能(例如操作系统提供的网络报文采集功能)和专用的攻击工具(例如专门编写的恶意代码),以手工或自动的方式来实施攻击过程。概括地看,网络攻击活动主要是基于安全漏洞的发现与利用。但是,随着针对的网络安全目标的不同,网络攻击会呈现不同的形式,并使用不同的技术。针对网络拥有性的攻击以渗透和控制为主,木马和僵尸网络都是其中的典型代表;针对网络可用性的攻击以服务失效为主,例如 DDoS 攻击;而中间人攻击则可同时针对网络真实性和保密性。

与网络防御相关的内容可以概括为多项单元技术,用以满足实现网络安全目标的需要。常用的网络防御相关技术包括密码技术、完整性控制技术、实体鉴别技术、访问控制技术、网络入侵检测与阻断技术等。**密码技术**提供各种对称与非对称加密方法以满足不同场合下网络保密性的要求。**完整性控制技术**提供各种信息摘录算法和**数字签名**方法,以满足网络完整性的要求。**实体鉴别技术**重点支持网络环境中的**身份认证**和源点鉴别,以满足网络真实性的要求;同时还包括无否认、匿名通信等与鉴别相关的技术,提供公平性判定及隐私保护等服务。网络环境中的**访问控制技术**同时涉及多个网络安全目标,除涵盖操作系统的各种传统的访问控制技术外,还涉及传输数据的访问限制,主要体现为虚拟专网技术的应用,例如基于 SSL/TLS 或 IPsec 的加密通路。网络入侵检测与阻断技术重点是满足网络可用性和网络拥有性的需要,包含的内容包括**网络入侵检测**技术,**防火墙**技术,恶意代码捕捉、检测与拦截技术等。

网络安全管理是网络安全目标达成的保障手段。相关的内容包括密钥管理技术、网络安全风险评估技术、对网络攻击的应急响应技术等。密钥管理技术涵盖对称和非对称密钥的生成、分配、使用、存储、更新和销毁等各个环节,口令管理也可以认为是包括在其中。网络安全风险评估主要涉及网络的安全规划和安全威胁发现两方面,网络安全管理员需要根据具体的需求(成本效益分析)确定网络的安全管理策略与安全功能设置,并根据具体的配置情况进行安全漏洞检测和关联性分析,以发现网络中存在的各种被攻击途径。由于网络的配置是可以动态变化的,因此这是一项经常性的工作,而不仅限于网络规划设计阶段。应急响应技术涉及的是在检测到网络攻击的前后对网络防御技术的使用,包括网络入侵检测系统的结果分析、网络安全设备的配置调整、灾备与恢复等。

网络安全的实现是一个涉及系统、用户和管理的复杂系统工程,需要有明确的安全目标和与之相适应的网络防御技术和安全管理技术来支持。整个网络的安全程度是用其最薄弱之处来衡量的,因此策略、技术、管理和教育缺一不可,需要有效均衡。

参考文献

龚俭, 吴桦, 杨望, 等. 计算机网络安全导论. 2 版. 南京: 东南大学出版社, 2007　(龚俭)

wangluo anquan chuanshu xieyi

网络安全传输协议(network security protocol)　一类用来保护网络中传输数据,使其免遭泄露、篡改和冒充等安全威胁的网络协议,又称**网络安全协议**。这些协议使用的最主要保护机制是数据加密技术以及基于加密技术的数据完整性保护技术和鉴别技术。

传统的数据传输协议(如 FTP、HTTP 等)和交互协议(如 Telnet 等),使用的是明文传输方式,这种无保护的传输方式不能满足敏感数据传输的需要,例如远程访问时口令的传输。随着**电子商务**、**电子政务**等网络应用的涌现与普及,数据传输保护成为必需的功能。人们或者对传统的传输协议进行了改进,增加相应的安全功能;或者重新设计新的安全协议或规程,作为现有传统传输方式的补充,以弥补相关安全功能的缺失。例如在网络层,补充了 IPsec 作为 IP 协议的安全规程;在运输层,增加了 SSL/TLS 作为新的安全协议;在应用层,增加了 SSH 作为对 Telnet 类应用的补充,增加了 HTTPS 作为对 HT-

TP 的增强。

VPN 在一定程度上也可以被视作是一种网络安全协议技术，特别是基于安全隧道技术的 VPN，例如 Point-to-Point Tunneling Protocol（PPTP）和 Layer 2 Tunneling Protocol（L2TP）等，它们也是通过对通信对象进行身份认证，对传输负载进行加密保护来实现数据的安全传输的。

参考文献

龚俭，吴桦，杨望. 计算机网络安全导论. 南京：东南大学出版社，2009

（龚俭）

wangluo anquan guanli

网络安全管理（network security management）　网络管理应用的一个分支，其根本目标是保证网络和系统的可用性（可生存性）。网络安全管理的物理范围是网络节点（site）或节点集合，即安全域。从网络安全管理的角度看，网络节点是一个拥有计算机或与网络有关资源的组织，并不是传输概念上的一个设备（计算机或路由器）。需保护的对象随网络而不同，但一般都会涉及硬件、软件、数据、人员、文档等方面。网络节点具有自己的管理政策，可具有嵌套结构。网络的安全管理有一个众所周知的原则，即保护的代价应小于被保护对象的代价。但是代价是一个综合的概念，它不仅包括经济上的损失，还包括信誉等其他方面的因素。

网络安全管理的内容可分为三个。

（1）对网络和用户的使用行为进行动态监测、审计和跟踪，以能够了解网络及其用户过去发生过和现在正在发生的行为，包括访问操作和相应的流量。

（2）对网络当前的安全状态作出正确和准确的评估，发现存在的安全问题和安全隐患，从而为安全管理员改进系统的安全性提供依据。

（3）保证安全管理政策能够得到贯彻和实施。这意味着网络安全管理系统不仅仅是一个观测工具，而且是一个控制工具，可以根据观测结果或管理员的要求对网络和用户的行为实施反馈，以保证系统的安全性。

网络安全管理涉及网络安全规划、网络安全管理机构、网络安全管理系统和网络安全教育等多个方面，要完成的任务包括需要标识要保护的对象，确定保护的手段，找出可能的威胁，实现具体的安全措施并争取有较好的性价比，了解网络的安全状态并能根据情况的变化重新评估和调整安全措施。

网络安全规划的确立要使得各项安全政策能够具有一致性（不是有强有弱），为一个共同的目标服务。网络安全规划的主要内容是进行网络的安全需求分析和风险分析，在此基础之上，确定网络的系统备份与恢复策略以及网络应急事件的处理规程。网络安全规划的结果是网络的安全政策，表现为所有用户和管理员都应该知晓和遵守的一组安全规则。注意要使网络安全政策在技术上是可行的，在管理上是可贯彻的。

网络安全管理机构负责日常的网络安全管理工作。由于网络安全管理是网络管理（或具体地说是网络运行管理）的一部分，因此网络安全管理机构往往也是网络管理机构。网络安全管理往往是按域进行的，域边界不一定与网络的物理边界吻合。例如校园网中某些信息管理系统所在的计算机系统构成一个独立的安全域，具有自己的安全政策和安全服务；而校园网的其他部分则可属于另一个具有不同安全政策的安全域。一般说来，一个网的最高安全域与最高管理域应是一致的，受这个网络所对应的组织机构边界的约束。

在互联网中，存在多个安全域，它们彼此之间可以协调，但不存在制约关系。因此传统的互联网中的各个接入网络一直奉行各自为政的安全管理策略。1988 年互联网蠕虫事件发生后，美国国防部的 DARPA 组织建立了计算机紧急响应工作组（Computer Emergency Response Team，CERT），CERT 协调中心（CERT/CC）设在卡内基梅隆大学的软件工程学院（CMU/SEI）。这是美国政府资助的联网系统安全性（Networked System Survivability，NSS）项目的一部分，CERT 的任务包括与有关部门合作，处理与互联网入网主机有关的安全事件；对互联网用户进行安全意识的教育和宣传；针对现有系统的安全问题开展研究。由于 CERT/CC 被注册了商标，因此其他的从事网络安全事件处理的类似机构称为计算机安全事件响应工作组（Computer Security Incident Response Team，CSIRT），它们负责实施、协调和响应管辖范围内各网络节点出现的安全事件。多数国家的学术网都建有自己的 CSIRT，向联网的教育机构提供宏观的安全管理指导和安全事件处理的技术支持，这些 CSIRT 都是公益性质的。政府部门的内部网和一些企业则根据自己的需求与技术力量来建立 CSIRT，这些 CSIRT 仅负责自己网络的安全，具有更多的行政权利，原则上不对外服务（参与技术协作或协同处理的除外），因此是非公益的。

网络安全管理系统是网络安全管理的支撑工具,通过在网络中部署所规划的安全服务与安全机制(独立设备或系统功能),网络安全政策得以实现。因此根据OSI安全体系结构,在逻辑上网络安全管理包括系统安全管理、安全服务管理和安全机制管理等三个方面;在物理上网络安全管理系统往往并非独立存在,而是分散在这些安全设备与安全功能中,或者是作为一个功能集成在网络管理系统中,因为网络的安全管理与网络的性能管理、配置管理、差错管理和计费管理是紧密联系和数据共享的。

网络安全教育是网络安全管理中的重要内容,需要对网络中的所有用户和管理员不断进行安全教育,以提高安全意识,保证安全政策的落实和发挥作用。

参考文献

龚俭, 吴桦, 杨望. 计算机网络安全导论. 2版. 南京:东南大学出版社, 2007 (龚俭)

wangluo anquan pinggu

网络安全评估(network security assessment) 指依靠各种管理和技术手段对系统进行检测,找出可能存在的安全隐患的过程,包括根据检测结果分析、评估系统的安全状况,根据评估结果,制定恰当的安全策略,为系统安全的设计提供参考依据。又常被称为安全漏洞评估(vulnerability assessment)。这里的系统可以是一个服务,也可以是一个网络上的计算机,还可以是整个计算机网络。

网络被攻击的最根本原因是因为联网的计算机系统存在可以被渗透(exploit)的安全漏洞。据美国计算机应急响应工作组协调中心CERT/CC统计,成功的Web攻击事件中有大约95%都是由于没有对已知的漏洞进行修补。美国SANS(System Administration, Network, and Security)协会和联邦调查局会定期公布最新的20个最危险的漏洞。大多数通过互联网对计算机系统进行的入侵都可以归结为没有对这20个漏洞进行修补,例如曾经造成很大影响的红色代码(Code Red)和尼姆达(Nimda)蠕虫病毒。因此,对网络上的计算机以及由若干主机组成的局域网进行安全评估以发现其中存在的安全漏洞是网络入侵防范中的重要环节。

对于特定的服务而言,安全评估与软件设计阶段以及软件测试阶段的故障分析有些类似,但又不完全一样。在软件设计阶段,主要目的是要努力避免产生安全漏洞。在软件测试阶段,主要目的是要找出可能存在的安全漏洞。而安全漏洞评估主要是检验目标软件是否具有已知的安全漏洞。尽管这三个阶段的主要目的不同,但可以使用类似的模型方法。

对于一个由若干计算机组成的网络而言,安全漏洞评估除了检验各个计算机主机是否具有安全漏洞脆弱性,还要找出各个主机联系在一起之后可能产生的新的安全漏洞。因此安全漏洞评估技术可以分为以下两类:基于单机的局部安全漏洞评估(简称局部评估)和基于网络系统的整体脆弱性评估(简称整体评估)。

安全漏洞评估分为两个部分:一部分是根据安全漏洞的特征在系统寻找对应的匹配,这部分可认为是直接的安全漏洞;另一部分则是寻找这些被发现的安全漏洞之间的关联关系以及系统配置之间可能导致安全漏洞的关联关系,这部分可认为是间接的安全漏洞,例如在自主访问控制机制中由访问权限的传递性所导致的潜通道。第一部分的工作与基于主机的入侵检测有些相像,因此比较简单,例如使用安全扫描器;而第二部分的工作则是安全漏洞评估的研究重点,因为这种关联关系分析通常需要计算复杂性较高的算法来支持,关联的完备性和实现的有效性是其中的难点。

在安全漏洞评估中,基于攻击图的评估方法是较具代表性的经典方法。该方法使用攻击图来描述系统安全漏洞之间的关联关系,具有简单直观的特点,而且对于图的操作有很多经典的算法,易于实现。攻击图技术把研究对象抽象为两个目标主体:目标网络和攻击者。它认为目标网络和攻击者之间存在博弈的关系,即目标网络努力保持自己在正常的状态空间转化,而攻击者总是试图使目标网络向“不期望”的状态转化。它首先以面向攻击的方式分别对目标网络建模和攻击者建模,然后根据二者之间的相互作用关系产生攻击图。由于该技术能够自动发现未知的系统安全漏洞以及安全漏洞之间的关系,从攻击的角度展示了攻击者利用网络内存在的安全漏洞进行渗透,逐步提升权限以最终实现入侵意图的过程。

攻击图技术主要包括三部分内容:目标模型构建,重点解决对评估对象的抽象描述,描述对象同时包括对被保护对象和攻击者;攻击图构建,根据网络状态和所发现的安全漏洞罗列出攻击者所有的攻击可能性;攻击图分析,根据攻击图计算各个攻击成功的可能性,并以此来评估网络的安全性。

参考文献

McNab C. Network security assessment. 2nd ed. O'Reilly Media, 2007　（龚俭）

wangluo anquan shenji

网络安全审计(network security audit)　对一个组织(例如一个企业)的网络进行人工或自动的定量安全评估的过程。人工评估包括了对组织员工进行安全策略的交流,对网络中的网络设备、主机和应用软件进行安全漏洞扫描和渗透攻击,评审操作系统和应用程序的访问控制策略以及分析网络中各类系统的物理访问权限。自动评估指利用各类计算机辅助程序自动输出审计结果,如系统审计日志生成工具、系统变更监视工具等。网络安全审计不仅仅是渗透测试,更重要的是审查该组织安全策略的制定和实施情况,是一个动态的过程。网络安全审计常见的对象包括:

(1) 网络体系结构与配置;
(2) 硬件防火墙与路由器配置;
(3) 用户鉴权与访问控制管理策略;
(4) 系统升级与补丁管理策略;
(5) 系统配置;
(6) 系统服务与应用程序配置;
(7) 防病毒软件管理策略;
(8) 机密数据加密与处理策略;
(9) 备份系统管理策略;
(10) 本地安全策略;
(11) 事件响应队伍的构成与能力;
(12) 物理安全。

网络安全审计的常见过程包括以下几个部分。

(1) 安全策略定义　网络安全审计的目标是一个组织对网络进行管理和使用的安全策略的实施情况,因此网络安全审计的第一步是确定该组织目前正在使用的安全策略。

(2) 审计准备　在进行审计之前,审计者必须先了解审计的对象,这些对象包括网络中的各类主机、设备、主机上安装的应用服务与应用程序以及网络使用者,然后确定审计需要检查的项目列表以及获取这些审计项目结论使用的工具和方法。

(3) 现场审计　针对准备审计的对象,使用工具或调查方法获取审计所需的数据。

(4) 审计总结　针对审计中所获取的各类数据,给出定量的安全评估结果。

网络安全审计是网络安全评估的一部分。网络安全审计重在根据检测项目列表给出每个审计项目的结论,而网络安全评估则需要根据网络安全审计给出的审计项目结果对网络整体的安全等级和弱点进行评价。

参考文献

1. Jackson C. Network security auditing. Cisco Press, 2010

2. 龚俭, 吴桦, 杨望. 计算机网络安全导论. 南京: 东南大学出版社, 2009

3. Hayes B. Conducting a security audit: An introductory overview. http://www.symantec.com

（杨望）

wangluo anquan weixie

网络安全威胁(network security threats)　网络系统及其元素可能受到的潜在攻击或危害。攻击是指对网络的所有非授权操作或脆弱性利用的行为,危害是指这些行为导致的影响网络正常运行的后果。行为实施者称为攻击者,它们可能来自网络外部或内部。狭义网络安全威胁专指对网络运行和服务的攻击或危害;广义网络安全威胁还包括对网络上传输或存储的信息以及接入用户的攻击或危害。网络攻击多种多样、不断变异。但从危害结果看,可大致分为路由破坏型、资源消耗型和信息窃取型三大类。前者如路由攻击、域名解析系统攻击等;中者如分布式拒绝服务攻击、恶意代码(病毒、僵死和木马等)传播等;后者如网络入侵、网络诱骗等。

路由攻击　指通过制造恶意路由信息导致数据包偏离正常路由的过程。大致分为两种:一种是通过各种攻击直接构造恶意路由表项,把数据包引导到攻击者指向的地方,从而达到窃听数据包、使数据包不可达、制造带宽瓶颈或消耗节点能量等攻击目的;另一种是伪装成合法路由器来欺骗其他节点或路由器,从而达到窃听数据包、或拒绝服务、或破坏正常路由的目的。如 ARP 欺骗、DNS 欺骗或会话劫持等造成的恶意路由。

域名解析系统攻击　也称 DNS 欺骗、DNS 劫持或 DNS 缓存污染。指在取得域名解析记录控制权情况下,把被查询域名对应的原 IP 地址解析为攻击者指定 IP 地址的过程。其直接后果是把用户引导到错误的互联网站点,达到向未授权邮件服务器发送邮件、单击付费或钓鱼恶意网页、强迫浏览广告,甚至窃取用户信息等目的。DNS 攻击一般采用三种方式,一是 DNS 缓存污染,即直接在 DNS 记录缓存

中填入攻击者的恶意 IP 解析地址;二是 DNS 劫持,攻击者首先嗅探查询客户端和 DNS 服务器的对话,再通过猜测或复制相同的请求 ID 号把恶意 IP 地址作为解析结果超前返回给查询用户,达到欺骗的目的。三是 DNS 重定向,攻击者将 DNS 名称查询重定向到自己可控制的恶意 DNS 服务器,然后把用户引导到攻击者所指向的 IP 地址。

分布式拒绝服务攻击　或称网络 DDoS 攻击,指利用系统设计或实现上的漏洞使网络目标系统崩溃或资源耗尽,导致无法提供正常服务的过程。攻击源大多来自受控网络僵尸,旨在消耗目标对象的 CPU、存储、带宽和时间等资源。主要攻击对象有操作系统攻击、网络协议攻击、链路带宽攻击,以及路由器、DNS 服务器、出口防火墙、用户认证系统等关键设施攻击。

恶意代码传播或植入　指各种病毒、蠕虫、僵尸、课件、键盘刺探、特洛伊木马、嗅探破解、漏洞扫描、垃圾邮件、逻辑炸弹、恶意脚本、恶意网页或恶意 ActiveX 控件等小程序在网络上的扩散并潜驻用户系统的过程。

非授权使用　指对网络资源实施与预定安全策略不一致的恶意操作,也称网络入侵。旨在对目标信息实施存取、处理或破坏,使目标系统不可靠或不可用。网络入侵获取信息的手段多种多样,可大致分为直接获取、利用获取和诱骗获取三类。前者一般通过远程口令攻击(其过程一般为寻觅账号、猜测口令、探测口令、破解口令、现场清理和预留后门等步骤)获得目标系统的口令或根权限,然后进行网络嗅探而窃听信息。中者一般利用缓冲区溢出、木马、僵尸、课件、键盘刺探、恶意脚本和恶意 ActiveX 控件等预先植入的恶意代码进行窃取与收集。后者则通过社会工程学问题,利用虚假电子邮件、网络钓鱼网站和虚假电子商务等诱使用户泄露其隐私,而实施欺骗与收集。当然,许多非授权使用往往综合利用以上各种手段形成复合攻击,更具有攻击力。

参考文献

1. Shirey R. RFC2828: Internet Security Glossary. 2000

2. Mirkovic J, Reiher P. A taxonomy of DDoS attack and DDoS defense mechanisms. ACM SIGCOMM Computer Communication Review, 2004: 39-53

(李芝棠)

wangluo biaozhunhua zuzhi

网络标准化组织(network standard institutions)　一类研究、制定和发布与网络相关的技术标准的权威机构。国际权威的网络标准化组织包括国际标准化组织(ISO)、国际电信联盟(ITU)、互联网工程组(IETF)、国际电气与电子工程师协会(IEEE)和万维网联盟(W3C)等。

网络标准化组织的主要职责是研究、制定和发布面向网络设备/网络系统相互连接的技术标准,包括网络接口相互连接类技术标准、网络协议相互连通类技术标准、网络服务相互操作类技术标准,这三类网络技术标准通常简称为网络互联、互通和互操作标准。首先,网络标准化组织关注于网络设备/网络系统相互连接的技术标准,既不涉及政治、经济、商业等非技术方面的内容,也不涉及网络设备/网络系统内部的实现技术。

不同的网络标准化组织对网络技术方面的知识产权制定了不同的处理规则。例如 IETF 采用了化解知识产权问题的处理模式,要求所有申请成为网络标准的技术都必须事先声明放弃已有的知识产权;ITU 采用了规避知识产权问题的处理模式,在发布网络技术标准时,明确说明哪些技术具有知识产权保护,使用网络技术标准过程的知识产权问题由标准的使用者与知识产权拥有者进行协商。

网络标准化组织是一类长期而稳定的专业技术机构。由于网络系统的构建、运行和维护是一个长期的、连续的过程,网络技术的研究、开发和应用也是一个不断发展和完善的过程。同样,网络技术标准化过程也是一个长期的、连续的过程。权威的网络标准化组织一般都是历史悠久、组织架构稳定、工作目标明确的机构。

另外还存在一些针对单项网络技术的国际标准化组织,它们通常是非官方的,其中最有代表性的是万维网联盟(W3C)。该组织创建于 1994 年,是 Web 技术领域最具权威和影响力的国际中立性技术标准机构。到目前为止,W3C 已发布了 200 多项影响深远的 Web 技术标准及实施指南,如广为业界采用的超文本标记语言(标准通用标记语言下的一个应用)、可扩展标记语言(标准通用标记语言下的一个子集)以及帮助残障人士有效获得 Web 内容的信息无障碍指南(WCAG)等,有效促进了 Web 技术的互相兼容。

从网络技术发展历史看,网络标准化组织在网络技术研究以及产业化过程中发挥了关键的作用。

网络技术的成熟和产业化的一个重要标志是,网络标准化组织已经提出了较为完整的网络技术标准,并且已经得到大部分网络设备生产厂商的支持。衡量一个网络标准化组织影响力的重要标准是该组织制定和发布的网络技术标准是否在全球得到了广泛的支持,以及这些网络技术标准在工业界是否得到了广泛的应用。　(沈苏彬)

wangluo celiang

网络测量(network measurement)　特定测量工具或系统支持下的一种网络状态和流量特性的感知过程,以支持网络管理员或用户对网络可用性进行评估,以及对网络故障或其他存在的问题进行诊断。网络测量活动大致可体现在网络性能测量、网络拓扑测量和网络安全测量等三个方面。网络性能测量的主要目的是检测信道以及网络整体的负载情况和传输质量情况,这是网络测量中最常见的应用。网络拓扑测量可用于发现网络的拓扑结构和路由结构。网络的拓扑结构测量在管理域内通常用以发现网络设备的在线情况,在管理域间通常用以发现网络的拓扑结构和 IP 地址配置与使用情况。网络路由结构测量观察网络路由的变化情况,主要目的是监测网络的互联结构、连通性及其稳定性和鲁棒性。网络安全测量的目的是实现对网络安全状态的了解,例如通过漏洞扫描发现管理域内被管对象的安全缺陷;通过网络入侵检测系统或蜜罐系统测量管理域内当前承受的网络攻击威胁的类型与频度等。

网络测量对网络状态和流量特性的感知要求是通过被称为网络测度的一些可测量变量来体现的,即网络的测量需求表达为一些确定的网络测度;网络的测量动作是这些网络测度的取值过程;网络的测量结果就是这些网络测度的值。网络测度分为标准的和自定义的两类。出于各种传输网络的网络运行管理的需要,ITU-T 和 IETF 等国际标准化组织定义了许多标准的关于网络服务质量的测度,为网络服务提供者和使用者对网络服务的质量评估提供统一的依据。例如 ITU-T 就电信服务质量提出了 Y.1221(基于 IP 网络的流量控制与拥塞控制,2002 年)、Y.1541(基于 IP 的网络服务性能目标,2002 年)和 Y.1291(在分组网中服务质量支持的体系结构框架,2004 年)等建议;IETF 的 IPPM 工作组针对互联网的性能、服务质量和可靠性等方面提出了一系列基础性的标准测度,供网络管理员、终端用户或第三方测试者使用。出于研究或特定的评估需求,研究者、各种网络服务供应商以及第三方测试者可以自行定义所需要的网络测度。自定义的网络测度可以从标准的网络测度派生,也可以网络报文或报文序列的内容独立定义,通常用于特定的网络性质检测或状态评估的需要。例如在评估对网络资源的使用情况时可以定义流量公平性测度,在评估网络性能时可以定义基于网络通路时延的网络距离测度。

从实现形式看,网络测量可分为主动测量和被动测量两种。主动测量通过往网络中注入测量报文序列,并观察这个报文序列的传输效果来得到测量结果;而被动测量则是通过直接观察网络流量的数量、类型以及内容来得到测量结果。主动测量可以根据需要产生相应的流量,使测量者得到所需的测度值。但是这种测量流量可能会对网络的正常流量形成干扰,进而影响测量精度。更重要的,出于网络安全的考虑,许多网络服务供应商会在其网络互联边界禁止主动测量常用的 ICMP 协议报文的穿越,从而影响主动测量任务的完成。被动测量的完成仅依赖对被测网络的流量观察,且对其无影响,因此不存在上述问题;但是也可能由于观察不到预期的网络流量而不能完成测量任务。因此,主动测量通常因需驱动,例如用于故障诊断;而被动测量则较适用于需要持续观测的场合,例如网络入侵检测,或者用于网络管理中的网络服务质量监测、用户流量统计、网络可用性及其变化趋势评估等。

无论是主动测量还是被动测量,都需要使用特定的测量工具来实施,这些测量工具可以是基于通用硬件平台开发的软件程序,或者是基于特殊硬件平台开发的测量系统,还可以是专用的测量设备。最常见的网络主动测量工具是内嵌在操作系统中的一些网络通路状态测试命令,例如 ping 和 traceroute,前者用于测量从测试点到指定宿点的传输通路的传输质量,包括往返时延和丢包率;后者用于发现从测试点到指定宿点的传输通路的拓扑结构,并给出通路中各点的传输时延。多数的主动测量使用单个测量报文即可实现测量目标,但某些情况下需要使用特殊构造的报文序列。例如著名的带宽测量工具 Pathchar 通过往传输通路中注入报文序列,并观察报文序列中报文间距的改变,可以推测出该通路的瓶颈带宽。

最常见的网络被动测量工具有 Tcpdump 和 Sniffer,两者均可用于采集信道中传输的报文(或其部分内容)并提供协议识别和内容解码功能,供测

量者分析报文内容使用。被动测量可以对每个报文都进行测量,也可以采用某种抽样模型对报文进行选择测量。

随着网络信道容量的逐步提升,使用通用计算机作为网络被动测量平台已不能满足报文采集和处理的性能需要,从 20 世纪 90 年代末开始出现基于 FPGA 或网络处理器等硬件平台的网络被动测量系统,这些测量系统的数据报文采集能力与网络信道容量的发展基本同步,可满足对现有互联网各种速率信道的报文采集或流记录采集的需要。

测量任务在一个测量点即可完成的称为单点测量,需要同时涉及一个以上测量点的称为协同测量。如果单点测量不能获得测度计算所需要的全部信息,则需要使用协同测量。例如在测量两个端点之间的单向传输时延时,需要在两端同时测量特定报文离开源点的时间和到达宿点的时间,从而可以推算出这个报文在信道中的传输时长。日常的网络性能测量和故障诊断等所使用的基本都是单域测量,即每个管理者只能在自己的管辖范围之内对基础设施进行测量分析和诊断。但是随着互联网规模的扩大和管理结构的日益复杂,对于需要持续进行的网络测量活动,例如网络服务质量监测和网络安全监测,要求建立起支持协同测量的网络测量基础设施。例如 CAIDA(The Cooperative Association for Internet Data Analysis)作为国际上最为著名的互联网测量合作组织,从 20 世纪 90 年代末开始通过设置在美国本土和全球各地的合作测量点,对互联网的拓扑结构和结构中传输通路的往返时延进行了持续的测量,对传输通路中的 IP 报文序列进行了持续的抽样测量,测量结果向全球公开发布,供研究者使用。

参考文献

1. 程光, 龚俭. 互联网流测量. 南京: 东南大学出版社, 2008

2. http://www.caida.org/home/ (龚俭)

wangluo ceshi

网络测试(network test) 一种根据网络标准,判定网络实现对网络标准符合程度而进行的实验活动。由于大部分网络标准都是软件实现的,因此网络测试借鉴了很多软件测试的概念,属于软件测试中的黑盒测试,即只通过观察被测试网络的输入输出行为对其进行判定。

网络测试的研究内容包括测试组织、测试方法、测试生成、测试表示、测试执行和判决、测试结果分析等多个方面。

网络测试的划分有很多不同的角度。从测试手段的角度划分,可以分成主动测试和被动测试;从测试目标的角度划分,可以分成一致性测试、互操作性测试和性能测试;从体系结构的角度划分,可以分成多层协议测试和单层协议测试等。 (尹霞)

wangluo chuliqi

网络处理器(network processor) 面向网络和通信应用的可编程**微处理器**,它是一种可编程的芯片,可用来处理网络包,处理速度可以高达每秒上百万个包。与通用处理器不同,网络处理器专用于网络和通信协议的处理。对每个网络包,网络处理器都可以进行复杂而灵活的处理,并且处理方式可以通过编程进行定制,这使它成为网络系统供应商开发包处理设备的一种简便易行的媒介。根据 ISO/OSI 模型的网络协议层次,网络处理器可细分为链路层处理器、网络层处理器及上层处理器等。在速率方面,根据 SONET/SDH 层次大致可分为 OC-12/STM-4、OC-48/STM-16、OC-192/STM-64 等几个层次,相应的通信速率大致为 622 Mb/s、2.5 Gb/s 以及 10 Gb/s 等。

早期的高速通信和网络处理器大多由专用大规模集成电路(ASIC)来实现。专用集成电路具有效率高、速度快、省电以及可靠等优点,适合用于成熟稳定的通信协议处理。其主要缺点有:①功能转移难,不易改动;②设计和投片成本高,不适用于小规模生产。而随着**因特网**的飞速发展,网络处理器带来了通信设备设计方式的革新。

网络处理器产品在 1999—2000 年开始出现,最早的产品有 C-port C5、Intel IXP1200、Sitera、Agere 等。这些网络处理器大多采用片上系统(SOC)技术,它们大多集成了一个或多个**精简指令集计算机**(RISC)的**中央处理器**(CPU)、高速片外存储器接口、用于表查找以及数据包分类和包队列缓存等专用硬件加速部件以及专用的通信接口的物理层或链路层(如以太网的 MAC)等。大多数产品被设计为应对基本的 IP 协议交换功能,目标的速率多为 OC-48/STM-16(2.4 Gb/s)。这些网络处理器的基本设计思想为利用运行在多个 CPU 上的软件来实现网络协议处理的基本控制流以及较为简单的数据处理功能,用集成片内的专门硬件单元来加速实现用软件所不易实现的实时处理功能(如 CRC 校验、高速查找 IP 路由表格、数据包流分类、队列和缓存、包交

换等)。此外,片上集成的专用链路层协议使其可方便地和系统中的相应物理层芯片连接,可减少外部逻辑电路或芯片的数量。由于此时的网络处理器的体系结构比较简单和理想化,在实际应用中,用户所需的一些复杂实时处理功能往往难以达到。

2001 年出现的网络处理器的代表为 PMC-Sierra 和 Broadcom 等公司生产的基于 MIPS 指令集的精简指令集微处理器。这些网络处理器除了多个高速的 CPU 单元以外,还集成了千兆以太网接口以及其他标准的通信系统总线接口(如 SPI 3、SPI 4.2 等),其目标是提供一个高性能的微处理器部件以支持网络系统中的控制层和数据层软件处理。这些处理器在处理网络路由协议和信令协议方面获得了广泛的应用。

2002 年出现的网络处理器主要代表为 Intel 公司的 Xscale 处理器。这些处理器集成了多个高速的 CPU 单元,其数量可以大于 64,其互联结构类似于传统的大规模并行处理(MPP)系统,其编程模式也与 MPP 系统类似,支持虚拟共享存储及消息传递编程模式。每个处理单元有私有的程序存储器和少量的局部数据存储器。它们通过互联网络、外部高速存储总线以及通信外设,与硬件加速单元交换数据和控制消息;这种高度并行的体系结构在单片上实现了单个处理器不能达到的极高处理速率(大于 20 000 MIPS)。

硬件结构

网络处理器通常集成多个高速的精简指令集微处理器以实现一个高度并行的系统,CPU 之间由高速内部总线或互联交换网络相连;CPU 之间还实现了缓存的一致性协议,以便于编程。网络处理器中还通常集成了硬件加速单元以针对网络处理中常用的校验、查表、队列处理及流分类等功能;网络处理器还包括了标准通信总线,以便和外部通信器件互连。此外,网络处理器还实现了多个外部高速静态和动态存储器总线,从而消除网络处理中常常遇到的存储带宽的瓶颈,常见的总线有 Rambus、DDR-SDRAM、DDR-SRAM、QDR-SRAM 以及 RLDRAM 等。

发展趋势

网络处理器经过 3 代的发展,已经用在各种网络硬件设备中。其可编程性带来的多方面优点是其他技术所不能取代的。此外,网络处理器还把传统通信系统的面向硬件的设计方式逐步转向面向软件的设计方式,这简化了系统研发过程,带来了代码复用的优点,使得系统开发的智力投资可不断积累,从而减少系统开发的长期费用。未来网络处理器的发展趋势如下:

(1) 单个 CPU 的性能不断提高。

(2) 片内处理器数量和并行度提高,互联网络的速率也相应提高,以保证系统性能的可扩展性。

(3) 开放式的指令集,合理方便的编程接口,并支持多机多线程。保证两代间的兼容性,以保护用户的软件开发投资。

(4) 开放式标准总线的支持,保证网络处理器和其他通信器件的方便互连。

(5) 高速高效的外部存储器总线,以保证内部算法不受存储器带宽瓶颈的影响。

(6) 根据不同的应用场合,合理地集成专门的加速部件及通信单元。

参考文献

1. Giladi R. Network processors: architecture, programming and implementation. Boston: Morgan Kaufmann, 2008

2. Lekkas P C. Network processors: architectures, protocols and platforms. New York: McGraw Hill, 2003

(廖恒 余宏亮)

wangluo cunchu guanli

网络存储管理(network storage management)

在单计算机系统存储管理的基础上,对网络存储系统中的各种硬件和软件资源进行配置、管理和优化,使之能够提供高效、稳定、可靠的存储服务。网络存储管理所涉及的管理任务通常会处理多种对象:存储硬件设备(如磁盘和**磁盘阵列**等),存储网络组件(如光纤和**以太网**互联设备等),存储服务器、存储系统软件、存储应用软件、数据和虚拟存储空间等。

存储管理软件一般除了有逻辑单元号(logical unit number, LUN)分配、设备发现、存储分区、LUN 映射等基本功能外,还具有数据保护、空间管理和存档、业务恢复、灾难恢复等高级功能。

然而,随着存储规模的不断增长,现代数据中心通常都配备有各种不同生产厂商所提供的存储产品,由于各个存储厂商所提供的存储管理软件都遵循各自不同的标准和协议,导致数据中心中许多存储设备相互不兼容,互操作性很差,增加了存储管理的难度。

针对异构多厂商存储设备的存储管理问题,存储网络工业协会(Storage Network Industry Associa-

tion, SNIA)把 DMTF(Distributed Management Task Force)组织提出的管理规范 CIM (Common Information Model)和软件设计的基于 Web 的企业管理(WBEM)倡议引入到存储管理领域,并得到了多家存储相关厂商的认可和支持,于2002 年提出了 SMI-S(Storage Management Initiative Specification, 也称 SMI 规范)存储管理接口标准,并于2007 年通过 ISO 认证,成为国际性的存储管理标准。SMI-S 形成了一种正规的、抽象的模型,该模型可以使被管理应用程序(如存储资源管理、设备管理和数据管理等)用来对存储资源进行标准化。SMI-S 的目标是,在存储设备和管理软件之间提供标准化的通信方式,使存储管理实现厂商无关性(vendor-neutral),从而提高管理效率、降低管理成本。SMI-S 为存储管理提供了一套统一的接口和协议,使得数据中心和用户能够在存储网络中轻松集成和管理来自多个厂商的产品,并能更加自由的选择存储厂商,从而提升了存储管理的灵活性,同时也降低了存储管理的复杂性。另一方面,SMI-S 为网络存储行业定义了一个全新、开放的开发模式,使存储厂商能够专注于存储产品功能的开发,而省去了异构和专有接口开发整合所需的技术支持,加速了产品开发的进程。随着 SMI-S 的完善和推广,越来越多的存储国际厂商(如 IBM、HP、EMC 等)的产品都已实现了对 SMI-S 标准的支持。

参考文献

IBM 存储管理. http://www.ibm.com/software/cn/tivoli/solution/storage/

(全超 舒继武)

wangluo cunchu xieyi

网络存储协议(network storage protocol)

用于网络存储系统中互联存储节点、服务器、客户机等的协议。典型的有光纤通道协议(FCP, fibre channel protocol)、互联网小型计算机系统接口协议(iSCSI, internet small computer system interface)、光纤通道以太网协议(FCoE, fibre channel over Ethernet)、InfiniBand 互联及 iSCSI 远程直接传输协议(iSER, extensions for RDMA)等。

光纤通道是为服务器高速连接多个存储设备而设计的,其数据传输率由 1 Gb/s 发展到 2 Gb/s、8 Gb/s,属 SCSI(参见**外存储设备接口**)标准接口之一,用于创建存储区域网络 SAN(参见**存储区域网**)。能满足高端工作站、服务器、海量存储子网络、外设间通过集线器、交换机和点对点连接进行双向、串行数据通信等系统对高数据传输率的要求。光纤通道可以采用铜轴电缆和光导纤维作为连接设备(大多采用光纤媒介,而传统的铜轴电缆如双绞线等则可以用于小规模的网络连接部署)。其主要特性是支持热插拔性、高速带宽、远程连接、连接设备数量多等。

iSCSI(Internet SCSI)是 2003 年互联网工程任务部(Internet engineering task force, IETF)制定的一项标准,用于将 SCSI 数据块映射成**以太网**数据包,并采用以太网协议传送 SCSI 命令、响应和数据。iSCSI 用来建立和管理 IP 存储设备、IP 网关或路由设备、主机和客户机等之间的相互连接,采用广泛使用的以太网来构建 IP 存储局域网 IP SAN,实现不同服务器共享存储资源,并可以在不停机状态下扩充存储容量。

FCoE 是 2007 年由 INCITS(国际信息技术标准委员会)的 T11 委员会(和 FC 标准制定是同一组织)开始制定的标准,2009 年 6 月标准完成(FC-BB-5)。FCoE 是将光纤通道映射到以太网,从而可以在以太网上传输**存储区域网**(SAN)数据。FCoE 面向的是 10 Gb/s 以太网,其技术标准规定可以使用任何速度的网卡,但需要网卡支持 802.3x PAUSE 机制。FCoE 基于 FC 模型而来,仍然使用 FSPF 和 WWN/FC ID 等 FC 的寻址与封装技术,只是在外层新增加了 FCoE 报头和电信级以太网(Ethernet)报头封装和相应的寻址动作,可以理解为类似 IP 和 Ethernet 的关系。

InfiniBand 是一个统一的互联结构,既可用于高性能计算节点的互连,也可以将**磁盘阵列**、SANs、服务器和集群服务器等进行互联。InfiniBand 主要用于构建企业数据中心,实现高的可靠性、可用性、可扩展性和高的性能。它可以在相对短的距离内提供高带宽、低延迟的传输,而且在单个或多个互联网络中支持冗余的 I/O 通道,因此能保持数据中心在局部故障时仍能运转。InfiniBand 的基本带宽是 2.5 Gb/s,即 InfiniBand 1.x,若采用全双工模式,带宽为 5 Gb/s。如果要获取比 InfiniBand 1.x 更多的带宽,只要增加更多缆线就行。截至 2012 年,InfiniBand 双工传输率已达到 80 Gb/s。InfiniBand 的扩展性非常高,在一个子网内可支持上万个节点,而每个网络中可有几千个子网,每个安装的系统中可以有多个网络结构。InfiniBand 交换机通过子网路由分组,InfiniBand 路由器将多个子网连接在一起。相对于以太网,InfiniBand 可以更加分散地进行管理,每个子网内有一个管理器,其在路由分组、映射网络

拓扑、在网络内提供多个链路、监视性能方面等起决定性的作用。子网管理器也能保证在特别通道内的带宽,并为不同优先权的数据流提供不同级别的服务。子网并不一定是一个单独的设备,它可以是内置于交换机的智能部件。iSER(iSCSI extensions for RDMA),是 IB SAN(InfiniBand SAN)的一种协议,其主要作用是把 iSCSI 协议的命令和数据通过 RDMA 的方式运行在 InfiniBand 网络上,作为 iSCSI RDMA 的存储协议 iSER 已被互联网工程任务部(IETF)标准化。

参考文献

1. RFC 3720-Internet Small Computer Systems Interface (iSCSI). http://www.ietf.org/rfc/rfc3720.txt, 2012

2. FC-BB-5. http://fcoe.com/09-056v5.pdf, 2012

3. iSER on InfiniBand Networks Introduction. https://www.research.ibm.com/haifa/satran/ips/iSER-in-an-IB-network-V9.pdf, 2012

(冯丹)

wangluo dayinji

网络打印机(network printer) 通过内置打印服务器,将打印机作为独立的设备接入**局域网**或**互联网**的打印机(参见**输出设备**)。网络打印机只需插入网线分配 IP 地址(参见 **Internet 地址**)就完成连接。它改变了传统打印机作为计算机附属设备的地位,使之成为网络中的一个独立节点,用来完成信息处理和打印输出任务,网络中的其他成员可以直接访问和使用该打印机,以网络的速度实现高速输出。

网络打印机应具有高速、高效的特点。作为部门或企业的打印设备,对打印的质量、打印噪声和打印成本会有较高要求。这些要求决定了目前网络打印机的打印部分均采用激光打印方式。网络部分目前大部分采用**千兆位以太网接口**。网络打印机的软件一般包括管理软件和监视软件。管理软件实现对打印机的全方位管理和控制,同时还可以通过网络及时进行升级。管理软件需要对打印机内部控制器、网络流量和打印队列实现有效管理,根据打印需求对网络连接性能进行优化,能在多种**操作系统**平台下使用,并适应多种网络环境。目前大部分的网络打印管理软件都是基于 Web 方式的,使用简单便捷。一般而言,管理软件供网络管理者和高级用户使用。监视软件供普通用户使用,用来查看打印任务,打印机的工作状态等信息。(韩承德)

wangluo dizhi fanyi

网络地址翻译(network address translation) 一种将局部私有(保留)地址转化为全局 IP 地址的转换技术,也称网络地址转换,简称 NAT,可用于解决全局 IP 地址不足以及减少来自网络外部的攻击,隐藏并保护内部网络等目的。一个内部网络只需使用少量全局 IP 地址即可实现网络内所有计算机与互联网的通信需求。

网络地址翻译主要是依据一个记录内部私有地址和全局地址的映射表,修改 IP 报文的源 IP 地址或目的 IP 地址等信息实现。网络地址翻译主要有静态地址转换和动态地址转换两种类型。静态地址转换是在映射表中,为内部地址和全局地址建立一个固定的一对一映射。这种方式主要针对内部网络的服务器,确保外部对服务器的正确访问。动态地址转换通过则建立一个全局地址池,从全局地址池中选择一个未使用的地址与内部地址进行转换。映射表项在连接建立时动态建立,在连接终止时全局地址被回收。有一种使用较广泛的动态地址转换类型称为端口地址转换,根据端口号和地址进行映射转换,通过利用传输控制协议或用户数据报协议端口号来唯一标识某台 IP 主机,允许多个内部地址同时共用一个全局地址。

由于 IPv4 到 IPv6 过渡技术的需求,IPv4 和 IPv6 网络地址的翻译技术和相关的协议翻译技术也已经形成 IETF 的标准,可参见 RFC6052 和 RFC6145。RFC6052 定义了从 IPv4 到 IPv6 的网络地址无状态翻译算法,称之为转换地址,也定义了 IPv6 到 IPv4 的网络地址无状态翻译算法,称为可译地址。此外,IPv6 到 IPv4 的网络地址也可以进行有状态翻译,与 IPv4 的 NAT 类似。

参考文献

1. Srisuresh P, et al. RFC 3022: Traditional IP network address translator (Traditional NAT). 2001

2. Bao C, et al. RFC6052: IPv6 addressing of IPv4/IPv6 translators. 2010

3. Li X, et al. RFC6145: IP/ICMP translation algorithm. 2011

(苏金树)

wangluo diaoyu

网络钓鱼(Phishing) 在电子通信中企图假冒可信实体而获取他人敏感信息的一种欺骗方法。据考证,网络钓鱼技术最早产生于 1987 年,“Phishing”

一词的首度使用在 1996 年，是英文单词“Phreaking”和“fishing”的拼合。也有说是“Phone” 和“fishing”的拼合，因为早期网络欺骗往往是通过电话进行的。

网络钓鱼虽然方式繁多、不断翻新，但万变不离其宗，其基本过程可归纳为鱼饵放置、过滤逃避、实体假冒和信息获取等四个阶段。其中，鱼饵放置和实体假冒是最为关键的环节。

所谓鱼饵是钓鱼者为诱使用户自愿链接到预先伪造的实体而精心部署的各种诱惑陷阱。例如，通过僵尸网络向大量僵尸主机发送看似来自可信组织或个人的欺骗性电子邮件，内容包括通过社会工程学技术制造的虚假紧急 “更新”“核实”等要求，进而诱使或愚弄用户直接填写其账号或密码等个人信息，或单击邮件中列出的 URL 链接（实为钓鱼网站或跨接地址）。有的通过手机短信诱骗用户单击某指定网站，导致木马程序自动下载到用户手机上。有的通过自动语音电话或网络电话（VoIP）要求用户拨打对方电话来验证其账号等信息，一旦拨通，自动语音系统将诱导用户写入敏感信息或偷录用户的每一次按键。有的则通过网络聊天或在线游戏，假冒管理员发来虚假中奖信息，愚弄用户进入钓鱼网站。还有的通过在公共场合建立 Wi-Fi 接入网络，伪造合法无线宽带提供商的注册页面诱骗用户输入账号和密码，或通过恶意软件先感染受害者计算机而后窃取信息。

钓鱼者通过预先设计好的、某个真实可信实体的仿制品，大多是远端的 Web 网站或某个自动语音服务器，甚至是一个守候热线的人（如通过电话的网络钓鱼）来实施钓鱼攻击。这些仿制品往往在外观上与在线组织、网络银行、网络零售商、网络支付商或信用卡公司等可信实体网站几乎一模一样的网页，甚至有自动生成虚假 Web 网页的所谓“超级垂钓者”软件存在。这类实体假冒的技巧在于 URL 的伪造或拨接。有的利用 URL 字符相近的视觉陷阱，例如把字母“l” 写成数字“1”，或少写或多写其中少数字符来蒙骗用户。有的通过修改主机的 HOSTS 文件映射、或污染 DNS 记录把用户直接导向伪造网页。有的用 JavaScript 命令背后偷换图像 URL 条上的合法地址，有的则利用真实 Web 网站的脚本漏洞，使点击的合法网站暗渡到伪造网站。

可从技术与社会两方面加强反钓鱼措施。技术方面：可使用带反钓鱼功能的互联网浏览器；给服务器安装高可信根的 EV SSL 证书或给网站代码签章信任标记，从而给用户以权威真实和安全可信的明确指示。社会方面：要建立相应响应组织和机制，开展相关立法、宣传和培训，以提高用户识别网络钓鱼的警觉性。

参考文献

诸葛建伟. 网络攻防技术与实践. 北京：电子工业出版社，2011

（李芝棠）

wangluo fuwu zhiliang

网络服务质量（quality of network services）

发送和接收信息的用户之间以及用户与传输信息的网络之间关于信息传输的质量约定。它包括用户要求和网络服务提供者的行为两个方面。用户要求指用户在网络上进行多媒体通信时所要求的服务类型以及相应的传输性能和质量等，服务提供者的行为则指网络针对某一类服务所能提供和达到的性能与质量。

网络服务质量不是网络中某个个体或元素的行为描述，它涉及用户与用户、用户与网络以及网络内部节点的整体行为。

服务质量控制涉及用户与用户之间、用户与网络的各个元素之间的协调管理、资源分配等问题。

网络中引入服务质量控制的主要原因是为了传输实时的多媒体信息流和合理利用网络资源。目前 Internet 上传输的信息主要以数据流为主，而且主要是采用先来先服务的尽力而为的方式。这就阻碍了语音及图像等信息在 Internet 上的实时传输。

服务质量控制需要解决下述几类问题：

（1）服务质量的分类与定义　OSI 基准（参考）模型对服务质量的类型和参数进行了规范化定义，针对 OSI 基准（参考）模型，给出了每层协议所必须具有的服务质量。Internet 协会（IETF）对服务质量的分类和参数给出了严格定义，制定了与服务质量相关的一系列提案和标准，包括集成服务与区分服务的服务质量控制模型、实现框架和有关参数以及相关的资源预约协议。

（2）准入控制　根据用户的服务质量要求和网络资源情况，控制使用网络资源的用户数量，从而保证正在使用网络资源的用户和享有高优先权的用户所需要的服务质量。

（3）资源调度算法　网络是由各种软、硬件资源组成的，这些资源包括带宽、处理机、缓冲区、存储设备以及各种软件资源。如何根据用户的服务质量要求来调度和分配资源是服务质量控制所要研究解决的重要问题之一。

(4) 基于服务质量的路由　Internet 的路由算法是以最短路径法为基础的,即路由器对要在 Internet 上互相通信的两台计算机计算它们之间的跳数,并以跳数最小的路径作为通信路径,但对于时延、跳变以及传输信息的种类等则未作考虑。服务质量控制机制要求路由器在进行路由计算时,按照用户的服务质量要求选择通信路径。

(5) 资源预约协议　为了保证用户获得所要求的服务质量,网络必须为用户预留出相应的资源,例如带宽和处理机处理时间等。资源预约协议被用来为用户预留资源。

(6) 服务质量协商与再协商　用户的服务质量要求有时很难与网络所能提供的服务质量一致。而且在大多数情况下,服务质量要求可以在一定范围内进行调整,例如每秒钟传输图形的帧数。服务质量控制机制提供用户和网络之间的协商机制,以及在信息传输过程中网络资源发生变化后的再协商机制。

参考文献

张尧学,等. 计算机网络与 Internet 教程. 2 版. 北京: 清华大学出版社, 2006　(张尧学　徐明伟)

wangluo gongcheng

网络工程(network engineering)　指应用计算机、电子通信、数学及管理等学科原理,构建计算机网、电话网、广播电视网等网络系统。数学用于构造网络模型与网络算法,电子通信学科和计算机学科用于设计制造网络节点及其连接,工程学科用于制定规范、评估成本,管理学科用于计划、资源、质量、成本等管理。

网络工程的目标是搭建和管理具有经济性、可用性、扩展性、安全的网络系统,满足信息传递的需求。经济性是指网络系统的搭建必须在社会经济和技术可支撑的范围内;可用性指网络向用户提供服务的能力,一般通过正常服务维持时间来衡量;扩展性是指网络系统能够通过已有结构、节点和链路的扩展满足新用户、新业务的需求,而不影响网络的可用性、安全性和经济性;安全性是指网络上传递的信息安全,涉及网络上信息的保密性、完整性、可用性、真实性和可控性。

实现上述目标所需要的步骤称为网络工程过程,主要包括设计过程、建设过程和维护过程,覆盖需求、设计、工程建设、确认以及维护等活动。需求和设计活动生成网络建设可行性研究文档;工程建设活动是将设计文档中的内容实体化,形成满足目标的网络系统;确认及维护活动是通过对网络的测试、配置和管理,为用户提供网络服务。

网络工程的原则

围绕网络设计、工程建设以及网络运营管理,应遵循以下几条基本原则。

第一条原则　选择合适的组网模式。组网模式是指选择合适的网络结构和网络规模,实现业务流程,提供可用服务,在网络设计过程中,传播的信息属性、资源需求、服务用户范围和地理跨度等多种因素是相互制约和影响的,需要综合权衡;还应充分考虑组网技术的发展以及软硬件设备的进步,保证选择的组网模式具有可用性、经济性和扩展性。

第二条原则　提供高质量的工程支撑。工程实践是网络从图纸转化为实际提供服务的物理网络的重要步骤,良好的工程管理制度、完善的工程规范和自动化高效的工程管理工具是搭建高质量网络的重要支撑。

第三条原则　重视网络运营的各种流程。复杂网络的运营是由各种流程驱动的,流程的闭环性、流程的自动化和流程的有效性直接影响网络的服务提供水平和资源利用效率,只有推动高效的流程管理,才能实现高效的网络工程管理。

网络工程的研究内容

包括网络组织模型、网络设计与实现方法、网络生命周期、网络工具、网络经济分析学等。

网络组织模型主要研究网络组网理论、网络的统计规律、网络分层和网络协议等内容。网络组网结构是从计算机图论等原理出发,结合网络性质、业务特点以及工程实践能力,连接各种网络节点的逻辑结构。典型的组网结构包括总线型网络、树状网络和网状网络等,总线型网络一般用于控制系统网络中;树状网络则一般用于汇聚型网络中;而网状网络主要应用于连接节点多、流量特征复杂的广域网络中。

统计方法是目前分析网络现象的主要工具,如通过泊松分布仿真网络流量,通过马尔可夫过程分析网络性能和节点可靠性等。近年来,学术界还开始研究二八现象、长尾理论等复杂网络的统计特征和规律。

网络设计与实现方法主要指网络设计和实现的全过程和步骤,主要研究内容包括网络节点设计实现、网络组织和覆盖、网络测量与分析以及网络安全等。

（1）网络节点的设计与实现　网络节点是网络中承担接入、中继、路由转发、业务服务等功能的网元，多个网络节点之间通过网络协议交互，实现OSI网络分层中的某一层或若干层的功能。网络节点的实现应综合考虑其功能和容量需求、解决方案的成熟程度以及商业可行性等多个因素，选择合适的架构和芯片。网络节点包括接入节点、传送节点、路由节点、存储节点以及服务节点等。

（2）网络连接与网络覆盖　是指将网络节点之间或网络节点和用户终端之间相连接，组成网络并提供服务的过程。网络连接主要考虑物理距离、容量需求；而网络覆盖则主要考虑覆盖范围、覆盖用户数以及提供的接入容量等。

（3）网络测量与性能分析　网络管理机构通过定义和规范性能指标和有效的测量手段，了解网络系统的服务性能和服务能力，不同类型的网络，其服务性能指标和测量方法均存在较大差异。

（4）网络安全　是指网络系统的硬件、软件及其系统中的数据受到保护，不因偶然的或者恶意的原因而遭受到破坏、更改、泄露，系统连续可靠正常地运行，网络服务不中断。网络安全应具有五个方面的特征：①保密性，信息不泄露给非授权用户、实体或过程，或供其利用的特性；②完整性，数据未经授权不能进行改变的特性。即信息在存储或传输过程中保持不被修改、不被破坏和丢失的特性；③可用性，可被授权实体访问并按需求使用的特性，即当需要时能否存取所需的信息。例如网络环境下拒绝服务、破坏网络和有关系统的正常运行等都属于对可用性的攻击；④可控性，对信息的传播及内容具有控制能力；⑤可审查性：出现安全问题时提供依据与手段。

网络工具是指用来支撑或辅助网络设计、网络建设以及网络维护管理的软硬件工具。使用这些工具可以帮助我们更好地完成网络生命周期内的各项活动，提升工作效率和网络服务质量。

（1）网络仿真工具　模拟网络现象、用户行为和网络流量的软件，在网络规划阶段辅助确定关键设计参数以及关键技术。

（2）网络分析工具　采集网络流量、网络性能指标等重要运行指标数据并进行分析的辅助工具，为下一步网络优化和扩容提供参考。

（3）网络测试工具　对网络设备以及整个网络进行测试的软件或硬件，主要为了检验网络设备的功能、性能以及网络提供特定服务的能力。

（4）网络维护管理工具　网络运行维护人员使用的工具，帮助运维人员配置业务数据、监控网络运行情况，快速发现和定位网络故障。

（5）工程管理工具　在建设网络时，为保障网络建设质量，支撑整个网络建设全过程的管理工具。

（韦乐平）

wangluo gongji

网络攻击（network attacks）　利用网络中存在的安全漏洞与缺陷，以破坏机密性、完整性和可用性等安全属性为目的，对信息系统资源所进行的攻击行为。

网络攻击可分为主动攻击与被动攻击两类。主动攻击试图改变目标信息系统的资源或者影响其运行，而被动攻击则仅仅从目标信息系统获取或利用信息，但不对信息系统资源和运行造成影响。网络攻击也可以根据发起位置分为内部攻击和外部攻击，内部攻击是由安全边界内部的实体所发起，而外部攻击则由安全边界外部的非授权用户或入侵者发起。

网络攻击的四种基本模式为截获、篡改、中断和伪造。

（1）**截获**　获取网络通信实体的隐私或机密信息，具体攻击技术为网络嗅探（sniffing）、通信推理（inference）等。截获是一种被动攻击方式，一般只是静默地监听网络通信流量，因此很难被通信实体所发觉。

（2）**篡改**　对网络通信实体的信息内容、程序或系统进行修改，使得通信方接受到虚假信息，或者程序和系统按非预期的方式执行，具体攻击技术为报文篡改（tampering）、渗透（exploitation）与误用（misuse）。

（3）**中断**　致使授权用户的正常网络通信与服务无法继续，具体攻击技术为**分布式拒绝服务**、系统损坏（corruption）等。

（4）**伪造**　假冒合法实体身份，欺骗网络通信对方以达到恶意目的，具体攻击技术为**网络欺骗**、冒充（masquerade）等。

篡改、中断与伪造这三种基本攻击模式都属于主动攻击的范畴，都需要攻击者主动改变信息系统的通信或运行过程，从而达成恶意目的。

上述四种基本攻击模式可以组合出许多高级网络攻击方式，例如**中间人攻击**（man in the middle）。中间人攻击实质上是对截获、篡改和伪造这三种基本模式的组合应用，位于网络通信双方实体中间的

攻击者首先通过截获监听双方全部通信信息，然后注入篡改后的消息，并对通信双方分别伪造身份，使其相信攻击者就是合法的通信对方，从而对通信会话进行任意操纵。会话劫持即是采用中间人攻击模式的一种具体网络攻击技术。

针对信息系统资源实施的完整网络攻击包括以下基本步骤：踩点（footprinting）、扫描（scanning）、查点（enumeration）、获取访问（gain access）、特权提升（escalating privilege）、拒绝服务（denial of service）、偷盗窃取（pilfering）、掩踪灭迹（covering tracks）和创建后门（creating backdoors），而这 9 个步骤又分为窥探设施、攻击目标、成功之后这三个阶段，如图 1 所示。

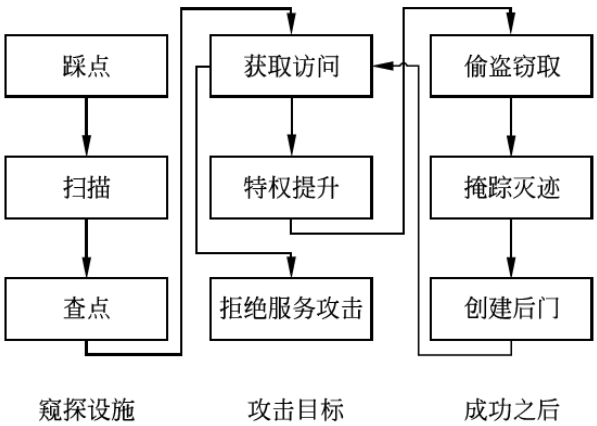


图 1　网络攻击基本步骤图

（1）踩点　是指攻击者通过对目标组织或个人进行有计划、有步骤的信息收集，从而了解攻击目标的网络环境和信息安全状况，得到攻击目标完整剖析图的技术过程。通过对完整剖析图的细致分析，攻击者将会从中寻找出攻击目标可能存在的薄弱环节，为进一步的攻击行动提供指引。

（2）扫描　基本目的是探测目标网络，以找出尽可能多的连接目标，然后再进一步探测获取系统类型、存在的安全弱点等信息，为进一步攻击选择恰当目标与通道提供支持。

（3）查点　在实施远程渗透攻击之前，针对已知弱点和识别出来的服务进行更加充分更具针对性的探查，来寻找真正可以攻击的入口，以及攻击过程中可能需要的关键数据。

（4）获取访问　收集到足够多信息之后，就可以尝试远程或本地攻击目标系统存在的弱点与漏洞，从而获得目标系统的访问权。

（5）特权提升　一般情况下通过网络攻击获取的初始访问都是受限用户账号权限，而攻击者进一步在目标系统上利用操作系统安全漏洞将受限用户账号提升至管理员用户账号权限，以取得对目标系统的完全控制权。

（6）拒绝服务攻击　有些恶意网络攻击者会利用分布式拒绝服务攻击造成网络中的目标系统变得无法访问，从而使得正常业务流程中断或受到影响。

（7）偷盗窃取　网络攻击者在取得目标系统的访问之后，根据其攻击目的不同，会从目标系统中窃取不同类型的敏感信息与数字资产，比如商业情报、国家秘密、个人隐私信息、网游娱乐中的虚拟资产等。

（8）掩踪灭迹　网络攻击者在目标系统与网络中会采用各种技术来隐藏攻击踪迹并清除行为记录，以防止攻击被发觉和躲避执法部门追查。

（9）创建后门　为了对攻陷的目标系统保持长期持久访问，攻击者通常还会在系统中安装一些特洛伊木马、后门软件或内核套件，来创建一个可以绕过正常认证机制的隐蔽访问后门。

由于地缘政治、地下经济利益等因素的驱动以及网络攻击技术的发展，互联网上的网络攻击已经变得更加多样、复杂和自动化，而与此同时，发动攻击所需的技巧和知识却在下降，这导致了互联网上的网络攻击变得非常普遍，所造成的危害和社会影响也逐渐扩大。恶意代码、垃圾邮件、网络欺诈等针对大众网民的普遍性安全威胁将长期存在和发展，而针对大型企业与组织的高级持续性威胁（advanced persistent threat，APT）也不断涌现，成为网络攻击威胁中的社会关注热点。网络防范技术在和网络攻击的对抗过程中也在进行着不断创新与发展。在信息安全基础理论与方法未取得突破性进展的情况下，网络攻击威胁将长期普遍存在，网络攻防技术的相互博弈也将持续进行。

参考文献

1. NIST. An introduction to computer security: The NIST handbook. October, 1995

2. NIST. Standards for security categorization of federal information and information systems. February, 2004

3. McClure S, Scambray J, Kurtz G. Hacking exposed. 6th ed. New York: McGraw-Hill Osborne Media, 2009.

4. 诸葛建伟. 网络攻防技术与实践. 北京：电子工业出版社，2011

（诸葛建伟）

wangluo guanli

网络管理(network management) 通过监视、分析、控制等管理手段,采用人工或/和自动的管理方式,对网络资源进行管理的一种服务,其目的是保障网络正常、可靠、经济和安全地运行。上面这一句对于网络管理的定性叙述,给出了理解网络管理的四个重要方面:网络管理手段、网络管理方式、网络管理对象(网络资源)和网络管理目的。

基本概念

网络管理手段 网络管理手段包括网络规划、网络部署、网络监视、网络分析、网络控制、网络优化和网络测量等,其中网络监视、网络分析和网络控制是常用的网络管理手段,通过网络监视、网络分析和网络控制这三个常用的网络管理手段,可以实现网络管理的基本闭环流程:获取网络运行状态(网络监视)、分析网络运行状态(网络分析)和实施对网络的控制(网络控制)。

网络管理方式 网络管理方式主要有两种,人工方式和自动方式。早期、初级、简单的网络管理采用的是人工方式,随着网络技术的不断发展,网络的规模越来越大、网络结构越来越复杂、网络上的应用越来越多,采用人工方式已经不能进行网络管理,必须采用自动方式进行网络管理。采用自动方式进行网络管理的系统称为网络管理系统。

网络管理对象 网络管理对象是网络资源。从网络管理对象的角度来看,网络资源主要有三类,即网元设备、网络和网络应用。网元设备是单个节点设备,如通信设备、网络设备、计算设备、存储设备等。各种网元设备按照一定的方法建立相应的联系,这种联系实际上描述了各种网元设备之间的关系,这种关系就是网络。网络应用一般是驻留在网元设备上的应用软件。

网络管理目的 网络管理的目的是保障网络正常、可靠、经济和安全地运行。

网络管理功能 网络管理是对网络资源进行管理的一种服务,为了提供这种服务,就需要相应的网络管理功能。

常用的网络管理功能分为五个功能域:故障管理(fault management)、配置管理(configuration management)、计费管理(accounting management)、性能管理(performance management)和安全管理(security management),常用这 5 个功能域的首字母缩写词 FCAPS 来表示,FCAPS 的进一步介绍参见**网络管理功能**。

网络管理系统 网络管理是对网络资源进行管理的一种服务,为了提供这种服务,就需要相应的网络管理系统。网络管理系统是驻留在计算机系统上的应用软件系统,网络管理系统主要由 4 部分组成:网络管理应用、网络管理接口处理、网络管理数据库和网络管理人机界面。

网络管理体系结构 网络管理体系结构主要有 4 部分的内容:网络管理功能体系结构、网络管理物理体系结构、网络管理功能体系结构和网络管理物理体系结构之间的关系、网络管理物理体系结构中组件之间的关系。

参考文献

1. 谢希仁. 计算机网络. 5 版. 北京: 电子工业出版社, 2007

2. 孟洛明, 亓峰. 现代网络管理技术. 北京: 北京邮电大学出版社, 1999(2002 再版) (孟洛明)

wangluo guanli biaozhun

网络管理标准(network management standard) 对通信和计算机网络进行计算机辅助管理时所应遵循的规范和准则。随着网络规模不断扩大,网络应用越来越广泛,联网的设备种类和数量也在不断增多,这就造成网络的结构越来越复杂,因此必然要依靠自动化、智能化的网络管理系统来辅助人们进行运营、维护与管理。一个大型复杂的网络中具有不同类型的软、硬件,来自于不同的供应商,要进行统一的管理,就要遵循事先约定好的、统一的标准。

网络管理标准涉及管理体系结构、网络管理协议和管理信息库等方面内容。根据制定标准的组织不同,主要有三种不同的标准系列:

(1) 国际标准化组织(ISO)所制定的网络管理标准,其中 ISO 7498-4(Management Framework, Part 4 of Information Processing Systems—Open Systems Interconnection, Basic Reference Model)定义了开放系统互连的管理框架,而相应的协议标准为公共管理信息协议—ISO 9595(Common Management Information Protocol, CMIP)和公共管理信息服务—ISO 9596(Common Management Information Services, CMIS)。

(2) 国际电信联盟电信标准化部(ITU-T)所制定的标准系列,由于 CMIP 和 CMIS 是 ISO 与 ITU-T 联合制定的网络标准,ITU-T 对应的标准号分别为 X.710 和 X.720,此外,ITU-T 致力于开发和制定电

信网的网络管理标准,引进了一些 OSI 管理思想,形成了较为完善的电信网络管理推荐标准,即 TMN (Telecommunications Management Network) 网络管理。

(3) 互联网工程任务组(IETF)面向数据网和计算机网提出了基于 SNMP(Simple Network Management Protocol)的网络管理体系,该 SNMP 管理解决方案是在考虑迁移 CMIP/CMIS 至 TCP/IP 网络—CMOT(CMIP/CMIS On TCP/IP)和改进简单网关管理协议(Simple Gateway Management Protocol, SGMP)之后而提出的。SNMP 在实际应用环境中得到了检验和发展,1990 年 IETF 正式公布了 SNMPv1 管理框架,由 RFC 1155、RFC 1212 和 RFC 1157 组成,其后推出的 SNMPv2 管理框架由 RFC 1902—RFC 1907 等标准组成,SNMPv3 管理框架则由 RFC 3410—RFC 3415 等标准组成。目前 SNMP 已成为网络管理领域中事实上的工业标准,被广泛支持和应用。

参考文献

Sean Harnedy. 简单网络管理协议教程. 2 版. 胡谷雨,等译. 北京: 电子工业出版社, 1999

(马皓)

wangluo guanli fenlei

网络管理分类(classification of network management)　网络管理通过监测、控制和记录网络资源的性能和使用情况,给网络管理员提供自动化监视和配置网络设备和资源的手段,以维护网络的正常运行。网络管理采用管理者-代理模型。管理者可以是工作站、微机、服务器等,它发出管理操作的指令和接收来自代理的信息。代理则位于被管理设备的内部,把来自管理者的命令或信息请求转换为本设备特有的指令,完成管理者的指示,或返回它所在设备的信息。代理也可以把自己系统中发生的事件主动通知给管理者。

网络管理协议定义了网络管理者与被管代理之间的通信协议及信息交换方式,管理信息库(MIB)定义了管理者通过管理协议可以获取或者配置的信息。简单网络管理协议(SNMP)是 IETF 定义的对互联网进行管理的网络管理协议,包括三个版本,现在的最新版本是 SNMPv3,该版本在安全性、功能、性能方面都有很大的改善。SNMP 协议是互联网中应用最广泛的网络管理协议。公共管理信息服务/公共管理信息协议(CMIS/CMIP)是 ISO 定义的网络管理协议,提供了一个完整的网络管理方案。由于其对系统处理能力要求过高,在互联网络领域未得到广泛的应用。

按照网络管理系统的组织结构,网络管理可以分为集中式网络管理和分布式网络管理。

集中式网络管理(centralized network management)由一个中心管理系统负责对整个网络进行统一控制和管理,所有的管理功能和管理信息都集中在这里。它一般位于网络系统的主干或接近主干的位置。网络管理的过程通常包括数据采集和数据处理,然后提交给网络管理员;它可能还包括数据分析并提供解决方案,甚至可能不打扰管理员而自动处理一些情况,或者生成管理员需要的报告。网络管理员可以通过这个中心管理系统进行网络管理,并监视和控制网络资源的使用情况。

传统的网络管理系统采用集中式管理模式,这种模式适合于较小规模的计算机网络的管理。通用网络管理软件根据标准的网络管理协议开发,形成通用的网络管理平台。这些网络管理平台提供友好的管理员操作界面,并提供第三方软件集成接口。设备厂商也会针对其设备开发专门的网络管理软件。

在集中式网络管理中,由于全部网络管理功能和管理信息都集中在一个系统上,它具有安装部署以及配置管理简单方便的优点。但是它也具有明显的缺点:管理系统需要和分布在网络中的每个被管对象通信,其流量会影响网络的性能;如果被管对象和中心管理系统之间的连接发生故障,那么,对这个资源的管理就不能进行;如果中心管理系统发生故障,就会造成整个网络管理功能的瘫痪。

分布式网络管理(distributed network management)将网络管理的功能分布在多个子管理系统中,通过多个管理子系统的协作共同完成网络管理功能。分布式网络管理中的各个子系统可以采用相似的技术实现,或是异构的网络管理系统;它们之间可以采用分级式的管理结构,也可以采用完全分布式的管理结构。

在分级式的结构中,底层系统实现部分管理功能,并把处理后的信息传递给高层的系统,经高层系统集中后实现完整的网络管理功能。在完全分布式结构中,各个子管理系统是对等的。

异构系统之间的通信和信息交换使用的主要技术有:公共对象请求代理体系结构(CORBA)和 Web Service。基于 CORBA 的网络管理通过在异构的系统上添加中间件,来对外实现统一的接口,实现异构

系统之间的互操作。基于 Web Service 的网络管理把各个子管理系统的功能包装成结构良好的服务并进行发布,通过服务调用实现子系统的协作。这种方式具有良好的可扩展性,被广泛应用于构建分布式网络管理系统。

分布式网络管理可以提高网络管理的性能、可扩展性、可靠性,以及配置及部署的灵活性。随着网络规模的扩大,网络复杂性的增强,分布式网络管理成为网络管理的发展方向。

参考文献

1. 杨家海, 任宪坤, 王沛瑜. 网络管理原理与实现技术. 北京: 清华大学出版社, 2000

2. Commer D E. Automated network management systems—Current and future capabilities. Englewood Cliffs, NJ: Prentice Hall (北京:机械工业出版社影印版), 2006

3. Subramanian M. Network management: principles and practice. Boston, MA: Addison-Wesley, 2000.

4. Hegering H-G, Abeck S, Neumair B. Integrated management of networked systems—concepts, architectures, and their operational application. Waltham, MA: Morgan Kaufmann Publishers, 1999 (杨家海)

wangluo guanli gongju

网络管理工具(network management tools)

指辅助完成故障管理、计费管理、配置管理、性能管理、安全管理五大管理功能的一系列管理工具,是实施网络管理的重要组成部分。网络管理工具通过监测和获取网络资源运行数据,采用自动、半自动或者人工数据处理,将网络运行状况呈现给网络管理人员,辅助管理人员维护网络正常运行。网络管理工具一般采用集中式结构,由一个中心管理系统负责对整个网络进行统一管理,所有的管理功能和管理信息集中于此。

依据所采用网络管理协议的不同,网络管理工具可划分为:可管理各厂商设备的通用网络管理工具和管理某个厂商设备的专用网络管理工具。通用网络管理工具基于标准网络管理协议开发,形成通用的网络管理平台,这些网络管理平台提供友好的管理员操作界面,通常提供第三方软件集成接口。专用网络管理工具采用厂商自定义的网络管理协议,目标是有效管理本厂商生产的各类设备。

比较通用网络管理工具和专用网络管理工具,前者具有通用于各厂家设备、对各类运行设备提供规范、一致的管理的特点,缺点是管理功能受所采用网络管理协议的局限,可管理的内容和范围受厂商对于标准协议的支持程度的局限,无法实施深度管理。而专用网络管理工具由于采用厂商自定义协议,管理灵活度大,可根据产品需要定制管理策略。正因为此,各厂商的专用网络管理工具无论在管理内容、管理工具种类、所采用的管理协议等方面,差异较大,体现了各厂商在产品开发和网络管理工具方面的研发重点和技术成熟度。专用网络管理工具的缺点是只可管理本厂商生产的设备。

鉴于通用网络管理工具和专用网络管理工具的发展现状,网络运营单位通常需要同时采用两种工具实施网络管理。由通用网络管理工具完成设备基本信息、网络拓扑自动发现和图形展示、网络健康状况及运行状态展示、故障报警、初步的性能管理等基本功能。由专用网络管理工具完成系统版本管理、配置管理、性能管理、网络分析、一致性检测等附加功能。

除了以上两大类网络管理工具之外,日益发挥重大作用的是开源网络管理工具。这些工具通常为轻量级网络管理系统,影响较大的有 Cacti、Nagios、OpenNMS、Ntop 等。其中,Cacti 基于 SNMP 协议实现,通过 RRDTool 展示流量、CPU、负载等网络运行数据。相比其他同类管理工具,它具有可定制用户视图、使用灵活等特点。Nagios 是一个强大的网络服务监控工具,监控内容包括主机资源、服务运行环境情况、服务进程等。它提供集中告警,采用告警页面、E-mail、msn、短信等多种形式报告故障。OpenNMS 是一款企业级开源网络管理软件,采用跨平台管理,支持设备监控和应用监控,可进行网络节点自动发现、故障监控以及故障事件管理,也可进行资产、报表以及系统管理,同时提供良好的第三方集成接口。Ntop 基于网络数据包嗅探,对流经网络的数据包进行处理,能够识别网络协议以及网络使用者,对网络中的每个 IP 通信情况进行详细分析,提供 Web 方式的控制界面和丰富的图形界面展示。

随着网络应用的不断丰富和普及,网络管理工具的覆盖范围不断扩大,功能上呈现细化的趋势。向下延展到光纤管理工具、端口管理工具等,向上细分为由各种应用需求衍生的管理工具,如用户身份管理、数据中心管理、深度网络分析、安全管理、无线管理、视频和内容分发管理、IT 资源管理、IT 业务管理等多个方面。 (陈萍)

wangluo guanli gongneng

网络管理功能（network management functions） 完成网络管理任务、保障网络安全稳定运行的直接手段。网络管理具有非常广泛的功能，包括网络管理过程中的很多方面。国际标准化组织（ISO）、国际电信联盟的电信标准部（ITU-T）、互联网工程任务组（IETF）等标准化组织公布了一系列网络管理的国际标准。在这些标准中，国际标准化组织的 ISO/IEC 7498-4 文档广为业界所接受，该文档为开放系统互连（OSI）的管理确定了五大功能，具体包括：故障管理、计费管理、配置管理、性能管理及安全管理。此外，网络管理还包括流程管理、资源管理、拓扑管理、客户管理、网络操作人员管理等辅助功能，因其具体实现都与网络实际条件有关，我们重点关注 OSI 网络管理标准中确定的五大基本功能。

故障管理（fault management） 故障管理包括故障检测、隔离以及纠正 OSI 环境中异常操作三个方面。故障会引起开放系统出问题而不能达成既定操作目标，这些故障有可能是持续的或短暂的。在开放系统的操作中，故障被描述成特定事件，比如错误。错误检测提供了识别故障的能力。故障管理包括以下功能：

（1）维护并检查错误日志。

（2）接受错误检测通知并采取相应行动。

（3）跟踪并识别错误。

（4）执行诊断测试序列。

（5）纠正错误。

计费管理（accounting management） 计费管理是在 OSI 环境中，根据资源的使用进行收费，同时确定使用这些资源的成本。计费管理包括以下功能：

（1）通知用户需承担的费用或消耗的资源。

（2）能够设定计费限额，使得费率表同资源的使用相关联。

（3）在实现特定通信目标时，能够将使用到的多种资源进行费用合并。

配置管理（configuration management） 配置管理针对开放系统进行识别和实施控制，从开放系统中收集数据并为其提供数据，以便于互连服务的准备、初始化和启动，为互连服务提供连续操作，以及终止互连服务。配置管理包括以下功能：

（1）设置控制开放系统常规操作的参数。

（2）把名称同被管理对象和被管理对象集合关联起来。

（3）初始化和关闭被管理对象。

（4）根据需求收集开放系统当前环境的信息。

（5）获得开放系统环境重大变化的通告。

（6）改变开放系统的配置。

性能管理（performance management） 性能管理可以评估 OSI 环境中资源的行为及通信活动的效能。性能管理包括以下功能：

（1）搜集统计信息。

（2）维护并检查系统状态的历史日志。

（3）确定系统在自然和人为环境下的性能。

（4）变换系统的操作模式以利于实施性能管理活动。

安全管理（security management） 安全管理的目的是支持安全策略的实施。安全管理包括以下功能：

（1）安全服务和机制的创建、删除与控制。

（2）安全相关信息的分发。

（3）安全相关事件的报告。

参考文献

1. ISO/IEC 7498-4—1989 信息处理系统 开放系统互连 基本参考模型 第 4 部分：管理框架

2. 孟洛明. 现代网络管理技术. 修订版. 北京：北京邮电大学出版社，2001

3. 李文璟，王智立. 网络管理原理及技术. 北京：人民邮电出版社，2008

（张蓓）

wangluo guanli tixi jiegou

网络管理体系结构（network management architecture） 从功能和物理的角度描述网络管理系统的组成及其之间的关系。从功能的角度描述网络管理系统的组成，称为网络管理功能体系结构；从物理的角度描述网络管理系统的组成，称为网络管理物理体系结构；网络管理系统组成之间关系主要有两类：网络管理功能体系结构和网络管理物理体系结构之间的关系、网络管理物理体系结构中组件之间的关系。

（1）网络管理功能体系结构 从功能的角度来讨论网络管理系统的组成，称为网络管理功能体系结构，网络管理功能体系结构表示的是网络管理系统的内部功能属性，如图 1 所示。

（2）网络管理物理体系结构 从物理的角度来讨论网络管理系统的组成，称为网络管理物理体系结构，网络管理物理体系结构表示的是网络管理系统的外部物理形态，是网络管理系统实现人员、使用

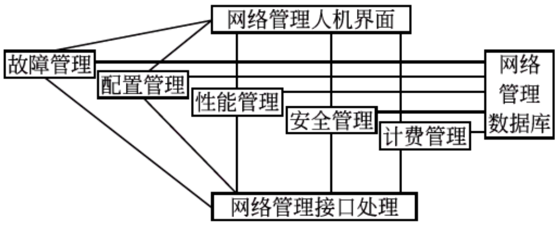


图 1　网络管理功能体系结构

人员可以感觉到的。在网络管理物理体系结构上的组件称为物理实体，一个物理实体就是一个驻留在计算机上的应用软件系统。

（3）网络管理功能体系结构和网络管理物理体系结构之间的关系　网络管理功能体系结构和网络管理物理体系结构之间的关系就是上面介绍网络管理物理体系结构时，介绍的功能实体和物理实体之间的映射关系。在网络管理功能体系结构的细化过程中，会得到一些功能相同的功能实体，可以将这些功能实体设计为标准的功能实体，这些标准的功能实体映射为物理实体的组成部分时，就成为可以重用的软件单元。

（4）网络管理物理体系结构中组件之间的关系　网络管理物理体系结构中的组件是物理实体，因此，网络管理物理体系结构中组件之间的关系就是物理实体之间的关系，物理实体之间的关系称为接口。

参考文献

孟洛明，亓峰. 现代网络管理技术. 北京：北京邮电大学出版社，1999（2002 再版）　（孟洛明）

wangluo guihua

网络规划（network planning）　一种根据网络需求，确定网络建设目标、主要技术经济指标的活动。网络服务（网络应用）是基于网络提供的，当有新的网络服务需要网络支撑，或者现有的网络服务发展，产生新的网络建设需求时，就需要进行网络规划。网络规划的结果（如网络的主要技术经济指标）用于指导网络工程中的下一个阶段，即网络设计。根据网络规划的对象网络不同，网络规划分为两种，一种是新建网络的网络规划，另一种是已有网络扩容时的网络规划。网络规划通常分为以下 4 个阶段。

（1）调查网络现状　调查网络现状是在已有网络扩容时，网络规划才有的一个阶段，如果是新建网络的网络规划，没有这个阶段。调查网络现状主要是收集网络结构、网络覆盖、网络性能、网络安全等方面的内容。网络结构主要包括网络体系结构、网络规模、编址和命名等。网络覆盖主要包括地理覆盖范围、覆盖用户情况、提供的接入容量等。网络性能主要包括容量、吞吐量、正确率（如丢包率、误码率）、效率、延迟、抖动和响应时间等。

调查网络现状主要是收集网络结构、网络覆盖、网络性能、网络安全等方面在网络设计时确定的设计数据和目前网络运行时的实际数据，对设计数据和实际数据之间的差异进行分析，并给出建议。

调查网络现状的主要方法有两种，即基于网络管理系统和人工调查。网络覆盖、网络性能、网络安全等方面的数据，一般的网络管理系统都进行采集、存储和分析，可以从网络管理系统中得到目前网络运行时的这些实际数据。人工调查主要是收集设计数据，采用的方法主要是查阅设计文件；网络结构的实际数据一般也是人工调查收集。

（2）分析网络需求　分析网络需求是根据网络服务来确定网络需求，网络服务是基于网络提供的，网络服务驱动网络规划，一般情况下，用户需求给出的是网络服务的需求，包括网络服务的种类、数量、服务性能和服务能力等，在分析网络需求阶段，要把服务的需求映射为网络结构、网络覆盖、网络性能、网络安全等网络方面的需求。

（3）确定建设目标　确定建设目标是根据技术方面的需求、经济方面的需求、资源方面的需求，在技术、经济、资源方面进行综合考虑，确定网络的建设目标。

在分析网络需求阶段确定的网络需求，主要是技术方面的需求，一般情况下，不能将这些技术方面的需求，直接映射为网络的技术指标，要综合考虑经济、资源方面的约束条件进行折中选择。

经济方面的需求主要是财务方面的投入和产出方面的需求，和分析技术方面的需求类似，在分析经济方面的需求时，也不能简单地将需求确定为投入最小，或产出最大，也要和技术、资源方面的约束条件进行折中选择。

资源方面的需求主要是号码（或地址）、频率、轨道空间（如果需要卫星通信，需要考虑卫星轨道方面的需求）。

（4）制定建设计划　在制定建设计划阶段，要确定网络体系结构、编址和命名等，要制定分期的建设计划，在每一期的建设计划中，要给出网络规模、网络覆盖、网络性能等技术方面的指标、经济方面的

指标(投入和产出)和所需的资源。制定的每一期的建设计划,用于指导该期的网络设计。(孟洛明)

wangluo hucaozuoxing ceshi

网络互操作性测试(network interoperability testing)　测试两个或更多的网络协议实现在网络环境中是否能够正确、平滑地交互,从而完成协议标准中规定的功能。互操作性测试的必要性主要体现在以下几个方面:

(1)一般来说,完备的一致性测试是不现实的,一致性测试集很难达到100%的测试覆盖;

(2)协议中往往包含一些可选功能,各个厂商可能还要进行一些扩展,对于可选功能的选取不同,可能会导致互操作失败;

(3)网络协议自身的版本也在不断更新,同一协议的不同版本实现存在差异。

互操作性测试已经被互联网工程任务组IETF和欧洲电信标准化协会ETSI等国际标准化组织广泛用于协议设计过程中。

网络互操作性测试采用最多的形式是选择互操作认可设备来与被测设备进行互操作性测试。认可设备是与被测设备连接互通、由其他厂商提供且已经被认为是合格的设备,它可能是终端设备、网络设备或者应用软件。认可设备可能是一个单独设备,也可能是若干设备的组合。网络互操作性测试和一致性测试的最大不同是在测试环境中用认可设备替代了测试仪表。

网络互操作性测试可以对系统实现的不同类型功能进行测试,包括协议规定的强制功能、协议规定的可选功能和协议未定义功能。网络互操作性测试发现的错误主要由两类因素所导致,一类是相应的协议规定不清楚使得实现出现差异,另一类是参数的设置发生冲突。

网络互操作性测试不对系统的性能、健壮性或可靠性作评估,也不对实现的一致性作测试。两个能够互操作的系统实现有可能与协议标准的定义并不一致。

网络互操作性测试的特征包括以下几点:被测设备与认可设备共同定义了测试的边界;被测设备与认可设备来自不同的厂商,或者至少是不同的生产线;测试基于提供给用户的功能来实施,这里的用户可以是人也可以是应用软件;测试在功能性接口上实施,例如人机接口、协议服务接口或应用编程接口。

参考文献

ISO/IEC. Information Technology, Open Systems Interconnection, Conformance Testing Methodology and Framework. ISO/IEC 9646. 1991

(徐明伟)

wangluo hulian

网络互联(internetworking)　将多个网络互相连接以实现在更大范围内的信息交换、资源共享和协同工作。互联网(Internet)就是由全球范围内成千上万个不同的网络互联而构成的网际网。**网络互联技术**包括**网络互联协议**和**网络互联设备**等方面。在网际网中,不同的网络应遵循相互认可的网络互联协议互连。信息传输的源和宿可以在不同的网络中,通过网际的**路由选择**实现穿越不同网络互联设备和中间网络的跨网络信息传输。(高传善)

wangluo hulian shebei

网络互联设备(internetworking equipment)　将网络连接起来的各种设备的总称,这些设备包括中继器、集线器、交换机、路由器、网关等。网络互联时,必须解决如下问题:在物理上如何把两种网络连接起来。一种网络如何与另一种网络实现互访与通信,如何解决它们之间协议方面的差别,如何处理速率与带宽的差别。解决这些问题的设备就是网络互联设备。

中继器(repeater)是工作于开放系统互连参考模型(OSI/RM)的第一层——物理层的一种网络互联设备。适用于完全相同的两类网络的互联,主要功能是通过对数据信号的重新发送或者转发,来扩大网络传输的距离。具有多个端口的中继器称为**集线器**(HUB)。

交换机(switch)是工作于开放系统互联参考模型(OSI/RM)的第二层——数据链路层的一种网络互联设备。是一种用于帧转发的网络设备,它可以



图1　互操作性测试

为接入交换机的任意两个网络节点提供独享的通路。最常见的交换机是以太网交换机。随着网络技术的发展,现在的某些交换机也带有路由选择功能,称为三层交换机。

路由器(router)是工作于开放系统互联参考模型(OSI/RM)的第三层——网络层的一种网络互联设备。路由器可以连接因特网中各局域网、广域网,实现具有相同或不同类型的网络的互联,它会根据网络协议自动选择和设定路由,以最佳路径,按先后顺序发送分组。

网关(gateway)是工作于开放系统互联参考模型(OSI/RM)的第四层——传输层以上(包括传输层)网络互联的设备。网关可以实现在采用不同体系结构或协议的网络之间进行互通时,提供翻译地址、转换协议、变换数据格式等功能。 (毕军)

wangluo jicheng fuwu

网络集成服务(network integrated services) 在 Internet 上同时提供传统的传输服务以及实时传输的数据、音频、视频等具有不同性能要求的传输服务。它实质上是从端到端的行为开始,到网络中各元素如何控制和实现这些行为,提供用户满意的服务质量的总称。

在 Internet 中,Internet 工程任务部(IETF)定义专门的规范和数据单元描述不同类型的集成服务。IETF 定义了尽力而为服务、保证型服务和控制负载型服务等三种类型。**尽力而为服务**指传统 Internet 使用的先来先服务模式。**保证型服务**要求用户清楚地描述应用需求,并指明实现机制。另外,支持保证型服务的网络路径中的各个元素,包括路由器和交换机等都必须支持保证型服务。否则,该服务将会因为其中某个元素带来的延迟而失败。**控制负载型服务**是一种端到端的控制行为,提供这种服务的网络使用户感到网络是在很轻负载或很大容量的条件下运行。用户的要求可能被延迟或得不到满足,但这种延迟和得不到满足都是用户可以忍耐的。在具体实现上,为保证控制负载型服务的条件得到满足,用户将首先给提供控制负载型服务的网络一个所需流量的估计值,然后,网络中各元素将验证自己是否具有足够的资源为用户提供服务。如果有的网络元素不具备相应的服务能力,则系统将其反馈给用户,并进行协商调整。

与集成服务相对应的服务质量控制机制是准入控制机构、资源预约协议以及相应的调度算法等。实现集成服务要求网络中各元素具有按照集成服务要求的资源控制能力和管理能力,这个要求较高。

参考文献

张尧学,等. 计算机网络与 Internet 教程. 2 版. 北京:清华大学出版社, 2006 (王勇 相士俊)

wangluo jisuan moshi

网络计算模式(network computing mode) 把地理位置上分布的多个计算资源通过计算机网络在逻辑上组织成一个集中的计算资源的方式。

网络计算模式从 20 世纪 70 年代开始,经历了从分时共享模式(主机体系结构)到资源共享模式(文件共享体系结构),再到**客户-服务器模式**的演变(参见**客户服务器计算**)。随之,网络计算的可用性、可伸缩性、互操作性及可扩展性也在逐渐增强。而网络计算模式的最重要的特点是它的可扩展性,它能够在不中断系统的情况下很快地适应系统的变化,例如系统负载的波动和某些节点的故障等。

网络计算模式的发展也促进了**网络运行环境**的发展,例如支持客户-服务器模式(参见**客户-服务器计算**)的一致通信环境和支持分布计算的**分布式计算环境**等。反之,网络运行环境的发展也产生了新的计算模式,如在万维网基础上发展起来的**浏览器-万维网-数据库模式**和**对等模式**等。

参考文献

胡道元. 计算机局域网. 4 版. 北京:清华大学出版社, 2010 (王勇 相士俊)

wangluo jiedian jiekou

网络节点接口(network node interface, NNI) 网络节点之间的接口或网络和网络之间的接口,一般为两个交换机之间的接口,与用户网络接口(UNI)一样,NNI 接口也定义了物理层、异步传送模式(ATM)层等各层的规范,以及信令等功能,但由于 NNI 接口关系到连接在网络中的路由选择问题,所以特别对路由选择方法做了说明。同样,NNI 接口也分为公网 NNI 和专用网中的 NNI(PNNI),公网 NNI 和 PNNI 的差别还是相当大的,如 ATM 网中的 NNI 的信令为 7 号信令体系的宽带综合业务数字网(ISDN)用户部分 B-ISUP,而 PNNI 则完全基于 UNI 接口,仍采用 UNI 的信令结构。

参考文献

1. 通信行业标准:自动交换光网络(ASON)技

术要求．第 1 部分：体系结构与总体要求．GB/T 21645．1—2008．2008

2．通信行业标准：数据通信名词术语．YD/T 1133—2001．2001 年（程时端　马严）

wangluo kongjian

网络空间（cyberspace）　由计算机、联网终端设备、路由器以及其他互联网基础设施组件之间通过链接创建的虚拟空间。它是人造的，可生长，可进化。

网络空间一词最初由科幻小说作家威廉·吉布森（William Gibson）于 1982 年在 *Omni* 杂志上发表的短篇小说以及随后的小说《神经漫游者（Neuromancer）》中首次使用。该书将网络空间描述为在一个充满人工智能生物的世界中，由计算机网络创造的空间。

Cyberspace 的含义随着计算机技术的发展而不断更新和扩展。Cyberspace 是控制论（cybernetics）和空间（space）两个词的组合。在 20 世纪 80 年代计算机开始普及时，前缀“cyber-”特指与计算机以及由它们组成的系统有关，进而特指一个由计算机组成的系统所提供给人们交互的“虚拟空间”，例如在线游戏和虚拟现实。随着互联网的发展，网络空间有时被直接用于指代**互联网**，包括在线游戏、聊天室、即时消息、博客、社交网站等网民交互的“场所”。网络空间更宽泛的概念还涉及现实生活中人与人之间的社交关系在网络空间中的投射。

“9·11 事件”之后，美国开始对互联网实施越来越严格的监管，并配套相应的组织机构和法律、法规，2001 年通过《爱国者法》，2003 年制定《网络空间安全国家战略》，2009 年美国军方成立网络司令部。美国政府将网络空间描述为“由互联网、电信网络、计算机系统以及关键产业中的嵌入式处理和控制器等信息技术基础设施组成的相互依存的网络，也用以指代信息以及人与人之间实现交互的虚拟环境”，并将网络空间的基础设施视为国家和国际安全体系、贸易网络、应急服务、基础通信以及其他公共和私人活动的基础，并通过相关立法加以保护。

从互联网的发展历史看，构成网络空间基础的数字基础设施的最初设计是基于网络互联互通和高效率的考虑，但是由于网络空间安全事件的频繁发生以及可能造成的巨大危害，网络空间的安全成为一个重要研究方向。为确保网络空间的可用性和安全性，各国基于自身的目标逐步建立各自的管理法规和体系，本质上无边界的、全球互联的网络空间正在形成有形的监管和控制区域。

参考文献

宫力．Cyberspace 的译名探讨．科技术语研究，2001，（1）

（徐明伟）

wangluo loudong

网络漏洞（network vulnerability）　被攻击者利用而使信息确保性降低的系统的弱点，也称脆弱性、缺陷或攻击接口。漏洞还有多种定义，例如，ISO 27005 定义其为被一个或多个威胁所利用的一种资产或一组资产的弱点，而资产是对组织、业务及其连续性以及支持组织关键任务的信息资源等有价值的任何东西。IETF RFC 2828 定义其为系统设计、实现或操作与管理中的缺陷或弱点，被利用后将破坏系统安全策略。而 The Open Group 则定义其为威胁能力超过抗威胁能力的概率。

根据涉及的系统资产类型，漏洞可划分为硬件、软件、网络、个人、网站或组织等多种类型。大众了解的漏洞一般指计算机软件的漏洞。引发软件漏洞的常见缺陷主要包括存储安全违规、输入合法性校验缺失、竞争条件冲突、特权混淆、特权扩大或用户接口错等。

漏洞与所在设备及所处环境相关。不同种类的软硬件设备、同种设备的不同版本、不同设备构成的不同系统以及同种系统的不同配置等都可能存在各自不同的安全漏洞。

漏洞还与时间紧密相关。新系统发布后，其漏洞会随着用户的深入使用而逐步暴露出来。而后，这些漏洞会被供应商补丁软件所修补或在升级版中纠正。当然，这种纠正或升级也可能产生新的漏洞。因此，旧漏洞的不断消失与新漏洞的不断出现是软件漏洞的一般过程，不能脱离具体时间和环境来讨论漏洞问题。

为了评估漏洞的安全风险程度，根据漏洞本身固有的特点（如是否可被远程利用）、危害影响、补丁推出速度及其生命周期等，相关组织和公司制定并开发了一个通用安全漏洞评估系统（common vulnerability scoring system，CVSS）。这是一个行业标准，利用它可对漏洞风险进行统一评分，这对不同评估系统的兼容、漏洞问题的交流、漏洞修复排序和及时把损失减到最小都是非常重要的。CVSS 已经广泛应用到主流的商业安全产品中。

为方便用户快速有效地鉴别、发现和修复软件产品的安全漏洞，根据 CVSS 标准建立有一个国际著名的通用漏洞与暴露 CVE（common vulnerabilities and exposures）字典，它是已知漏洞和暴露的标准化名称列表。此外，中国国家信息安全漏洞库（CNNVD）和国家信息安全漏洞共享平台（CNVD）也维护相应漏洞的信息。

“暴露”（exposures）是指对一些安全策略认为有问题，而对另一些安全策略却可被容忍的弱点。例如 finger 服务，既是本身业务所必须的，也可能为入侵者提供一些边缘信息或攻击尝试的可能性，但并不表明服务本身有安全问题。

寻找并利用计算机或网络漏洞的人被称为黑客。黑客早先更多指从事计算机犯罪的人，包括那些专注寻找漏洞、沉溺编写程序和图谋获取资料的“攻击者”和很少相关知识、却借用别人编写的程序就能侵入计算机系统的所谓“脚本小子”。现在也包括那些经常使用科学、工程或信息技术挑战现存网络秩序的人。他们对计算机和网络的内部运行机理及编程有足够理解。通常，根据其行为的目的或后果，可区分为侵入对方系统实施攻击破坏或窃取信息的黑帽黑客（black hat），也称骇客（cracker）；和专注寻错或破解，给系统提出警示或修补的白帽黑客（white hat）。前者具破坏性，后者具建设性。媒体常把“黑客”描述为网络违法者，大众则一般把“黑客”看作计算机罪犯。

参考文献

1. Peltier TR, Peltier J, Blackley JA. Managing a network vulnerability assessment. Auerbach Publications, 2003

2. http://cve.mitre.org/

3. http://www.first.org/cvss

（李芝棠）

wangluo loudong saomiao

网络漏洞扫描（network vulnerability scanner） 指基于漏洞数据库，通过网络对指定的网络、计算机、主机系统以及应用程序的安全脆弱性进行检测，发现可利用漏洞的一种安全检测（或渗透攻击）行为。网络漏洞扫描可能是安全审计人员对雇主网络的一次审计活动，也可能是黑客开展攻击前寻找目标漏洞的准备活动。网络漏洞扫描通过特定的软件工具实现。不同的网络漏洞扫描软件有不同的扫描目标和扫描方式。根据扫描目标的不同，网络漏洞扫描方式可以分为以下几种。

（1）端口扫描　扫描器通过向一台或多台主机的一个或多个服务端口发送请求报文，检测目标主机开启的端口以及端口上的网络服务。端口扫描是整个漏洞扫描工作的准备阶段，通过它，扫描者可以确定目标主机上开放的服务，从而进一步针对服务进行系统漏洞或应用漏洞扫描。

（2）系统漏洞扫描　扫描者模拟黑客的攻击手法对目标主机系统进行针对性的攻击尝试，如系统的越权漏洞攻击等。若模拟攻击成功，则表明目标主机系统存在安全漏洞。

（3）应用漏洞扫描　扫描者模拟黑客的攻击手法对目标应用系统如 Web 系统、数据库系统进行针对性的攻击尝试，如网站的注入攻击等。若模拟攻击成功，则表明目标应用系统存在安全漏洞。

除了目标的不同，根据扫描主机的方式还可以将网络漏洞扫描划分为遍历性漏洞扫描和针对性漏洞扫描。

（1）遍历性漏洞扫描没有明确的针对性，在一个大致的范围内对所有的主机作同样的扫描，以试图发现可能的攻击目标。遍历性扫描的常见做法包括顺序扫描、选择性随机扫描和拓扑扫描。

（2）针对性漏洞扫描先确定扫描的目标或目标范围，然后再进行扫描操作。因此扫描是有针对性的，针对扫描对象的不同，探测的漏洞类型会有所调整。常见的针对性扫描方法包括基于目标列表的扫描、基于路由的扫描和基于 DNS 的扫描。

常见的开源或免费扫描软件有 NMAP、NESSUS 和 MetaSploit，其中 NMAP 主要用于端口扫描，NESSUS 和 MetaSploit 用于系统漏洞扫描和通用的应用漏洞扫描。此外还有一些专用的应用漏洞扫描软件，例如针对 Web 应用程序漏洞的 APPSCAN。

参考文献

龚俭，吴桦，杨望. 计算机网络安全导论. 南京：东南大学出版社，2009

（杨望）

wangluo qipian

网络欺骗（network spoofing） 利用现有网络协议缺乏对通信双方的身份认证或认证机制不完善的安全缺陷，通过伪造数据包假冒通信方的身份所实施的一类网络攻击技术，是对网络通信系统真实性的违背。

网络欺骗的具体攻击技术

各个层次上的网络协议在设计时大多都没有严格的安全性考虑，因此普遍存在着认证机制方面的

安全缺陷，从而容易遭受网络欺骗攻击威胁。从链路层直至应用层，目前各个协议层次上存在的具体网络欺骗技术包括：

（1）MAC 地址欺骗　对联网设备网络接口的出厂 MAC 地址进行伪造的欺骗技术手段，以达到假冒其他主机进行非授权访问、隐藏攻击源等恶意目的。

（2）ARP 欺骗（ARP spoofing）　也被称为 ARP 下毒（ARP poisoning），指攻击者在有线以太网或无线网络上发送伪造 ARP 应答报文，对特定 IP 所对应的 MAC 地址进行假冒欺骗的攻击技术，目的是通信拦截或服务假冒。

（3）ICMP 路由重定向　攻击者伪装成路由器发送虚假的 ICMP 路由路径控制报文，使得受害主机选择攻击者指定路由路径，从而进行嗅探或假冒攻击的一种技术。

（4）IP 欺骗（IP spoofing）　攻击者伪造具有虚假源地址的 IP 数据包进行发送，以达到隐藏发送者身份、假冒其他计算机等目的。

（5）TCP RST 攻击　一种通过发送假冒通信方 IP 地址的伪造 TCP 重置数据包，干扰 TCP 通信连接造成通信中断的技术方法。

（6）DNS 欺骗（DNS spoofing）　攻击者通过冒充域名服务器、DNS 缓存下毒攻击等方式，对域名对应的 IP 地址进行假冒以达到通信拦截目的的欺骗技术。

（7）Email 欺骗（Email spoofing）　对电子邮件头部信息进行伪造，导致邮件看上去来源于某人或某个地方，而实际却不是真实发信人的欺骗技术。

（8）网络钓鱼（phishing）　利用欺骗性电子邮件和伪造 Web 站点实施的网络诈骗活动，被诱使访问的受骗者往往会泄露自己的私人资料，如信用卡号、银行卡账户、身份证号等内容。

网络欺骗的应对技术与措施

网络欺骗攻击主要利用了基础网络协议中存在的安全缺陷，安全研究领域针对这些缺陷已经提出了多种多样的安全增强机制，例如 RFC 2827 标准采用网络入站过滤机制来对抗 IP 源地址欺骗，RFC 4953 标准针对 TCP 欺骗攻击提出的多种传输层和网络层应对措施。但目前在实际网络中，这些应对措施尚未得到广泛部署，因此仍需要网络管理员和使用者采用一些最佳安全实践措施来安全配置与使用网络，避免遭受网络欺骗攻击的危害。

参考文献

1. Ferguson P, Senie D. Network ingress filtering: defeating denial of service attacks which employ IP source address spoofing. RFC 2827, 2000

2. Touch J. Defending TCP against spoofing attacks. RFC 2953, 2007

（诸葛建伟）

wangluo qufen fuwu

网络区分服务（network differential services，diffserv，DS）　除了提供传统的传输服务外，通过对数据分类以及利用应用本身的可适应性等对实时传输的数据、音频和视频等不同性能要求的应用提供可扩展的定性传输服务。

为了解决网络传输的服务质量控制问题（参见**网络服务质量**），Internet 工程任务部首先提出了集成服务模型。但要从现行的尽力而为传输模式升级到集成服务体系结构（参见**网络集成服务**）仍然存在许多困难。其中最主要的原因之一在于许多网络节点所采用的技术并不支持服务质量的实现，或缺乏合适的信令支持，从而削弱了网络提供端到端保证的能力。另一方面，由于集成服务体系结构的实施粒度比较细，所有的服务与控制都是针对数据流进行的，因此提供强服务保证的同时也带来了实现复杂、开销大和可伸缩性差等问题。

基于以上原因，研究人员尝试从另一个角度来考虑服务质量控制问题。例如，通过对数据分组进行分类以及利用应用本身的可适应性来满足各类服务的质量要求。由于只有小部分应用需要很强的服务质量保证，而这种保证可以通过提供足够高的带宽满足峰值流量来达到，因此仅做粗略的优先级分类就足以保护它们。这些想法构成了区分服务的基本思想。

区分服务体系结构基于比较简单的模型：当数据流进一个网络时，它的分组在边缘设备中被分类标记，从而可以将各个分组归入各种具有相同码值的分组集合。每种分组集合通过一个单一的 DS 码值来标识。在网络核心中，路由器根据分组的 DS 码值采用相应的“每跳行为（PHB）”来转发分组，不同的 PHB 使得分组在通过路由器时具有不同的处理优先级。PHB 被定义为“在某一个 DS 兼容节点上对通过一条链路的具有相同 DS 码值的分组集合所施加的外部可见的转发行为”，它是区分服务的核心功能。PHB 是网络节点在不同汇聚流之间分配缓存和带宽资源的具体手段，它主要通过缓存管

理和调度机制来实现。一组相关的 PHB 代表了区分网络的一类服务水平,通过为业务流标记 PHB 码值,就可以使这个业务流得到相应的服务。

区分服务功能由网络节点中许多功能元素实现,包括 PHB 集、分组分类功能、流量调节功能(测试、标记、整形和监控)等。

区分服务的显著特点是可伸缩性强,因而可以在大型网络中使用。可伸缩性强是通过将大量的复杂性工作放在边界设备中完成来达到的。边界设备降低业务流的速率并减少业务流的数目,同时使服务基于汇聚流而非基于每个微流的粒度,从而使核心路由器的工作仅限于转发汇聚业务流,这样做有利于以后的扩展。

参考文献

1. Blake S, Black D, Carlson M, et al. An architecture for differentiated services. Internet RFC 2475, Dec. 1998

2. Nichols K, Blake S, Baker F, et al. Definition of the differentiated services field (DS field) in the IPv4 and IPv6 headers. Internet RFC 2474. Dec. 1998

(王晓春 徐明伟)

wangluo ruqin fangfan

网络入侵防范(network intrusion prevention) 通过对网络流量的监视发现网络入侵行为(如扫描、服务器权限获取等),并通过主动手段拦截和阻止入侵行为的一种主动安全防御方法。网络入侵防范系统(network intrusion prevention system, NIPS)是网络入侵防范使用的一种安全工具,可基于用户定义的安全策略规则自动地对发现的入侵行为进行拦截和阻止。NIPS 可以是一套独立的系统,也可以是由网络入侵检测系统、防火墙系统以及路由器和交互机的安全模块联合组成的系统。不论哪一种系统组成方法,NIPS 需要同时具有入侵检测的功能与响应的功能。

网络入侵防范的结构

网络入侵防范的网络部署环境根据是否旁路流量可以分为两种形式,即被动和在线。

基于被动采集流量模式的 NIPS 由流量采集系统、检测节点和管理节点组成。流量采集系统通过交换机侦听或分光的方式被动采集流量副本,并将其发送给检测节点进行入侵检测。检测节点发现入侵后,将入侵警报发送给管理节点。管理节点基于响应规则指示其他网络安全设备对入侵流量进行拦截,并可向安全管理员发出警报。基于被动采集流量模式的 NIPS 对网络运行没有性能的影响,但由于无法直接干预流量,响应会有一定的滞后性。

基于在线采集流量模式的 NIPS 由入侵防范节点和管理节点组成,前者直接串联接入网络边界。当入侵防范节点检测到网络入侵后,基于响应规则直接对入侵流量进行拦截,同时将入侵警报发送给管理节点。基于在线采集流量模式的 NIPS 对网络入侵的响应速度更快,但由于其串联在通信链路上,对网络运行的性能可能会造成影响。

网络入侵防范的手段

(1)基于被动监测的入侵防范技术。在被动监测的情况下,入侵防范技术可以通过以下技术防范攻击:

中止 TCP 连接 入侵防范系统可以根据被动入侵检测系统提供警报获取入侵 TCP 连接信息,并向该 TCP 连接的两个终端分别发送 TCP RESET 报文来终止入侵的 TCP 连接。由于入侵防范系统可以获得 TCP 连接的终端 IP 地址、TCP 端口号、两端的序列号范围,因此构造的 TCP RESET 报文可以让连接的终端误认为对方希望结束这次的 TCP 连接,从而提前结束攻击的 TCP 连接。该方法的好处是对被保护的网络性能不会发生影响,缺点时由于构造和发送 TCP RESET 报文需要时间,因此可能在连接中止前攻击已经成功,或者入侵者可以修改系统协议栈,选择忽略 RESET 报文,从而让防范技术无效。此外这种技术只对有连接的 TCP 协议有效。

配置其他网络安全设备 在入侵检测系统发现入侵后,可以将入侵者的 IP 地址、攻击的端口等信息发送给路由器、交换机和防火墙等网络安全设备,通过配置这些网络设备的 ACL,将入侵的流量挡在被保护网络之外。常用的手段包括配置防火墙的黑名单以禁止攻击者进入被保护网络,或者在交换机上将被入侵的主机划入单独的 VLAN,阻止入侵者对网络的进一步访问。这种方法的好处是不需要为入侵防范再添置新的设备,只需要利用现有的网络安全设备或网络设备的安全功能即可;但缺点是这些设备的 ACL 大都只能基于 IP 地址和端口,如果入侵者的流量无法从 IP 地址和端口上和正常流量加以区分,则无法利用这些网络安全设备进行防范。

(2)基于在线监测的入侵防范技术。在在线监测的情况下,入侵防范技术可以通过以下技术防范攻击:

在线过滤 当入侵防范系统检测到入侵时,入

侵防范系统可以直接丢弃和拒绝入侵相关的报文和网络连接请求,起到类似防火墙的过滤功能。由于入侵防范系统对报文进行深度的检测,因此其过滤的精度远高于单独的防火墙。这种方法的好处是作用效果快,发现入侵时,可以立刻进行阻断,不会有配置和发送伪造报文的延迟,同时过滤的是经过深度检测的报文,相对于基于 IP 和端口的规则要更加精确。该方法的缺点是入侵防范系统本身会成为网络系统的瓶颈,如果攻击者针对入侵防范系统发送大量消耗系统资源的报文,则防范系统本身可能会因为性能不足而导致拒绝服务攻击的效果。

在线净化 除了丢弃入侵流量,有时入侵防范系统可以通过净化入侵流量来实现入侵防范。这种净化过程自动对报文的应用负载进行检查,并基于规则将所有可疑或禁止出现的应用负载特征进行替换,将可能存在入侵行为的应用负载变成无害的流量。最常见的净化是对 HTTP 协议的请求报文进行净化处理,将原始的请求编码成对服务器无害的请求形式,并替换掉其中可能逃逸 HTTP 服务器检查的部分,使得攻击无法在 HTTP 服务器上成功。

网络入侵防范的弱点

网络入侵防范是入侵防范技术中基于网络技术的一个分支,在实际使用时往往需要和其他入侵防范技术协同使用,例如主机入侵防范。网络入侵防范最大的弱点是对于加密的通信过程如基于 VPN、基于 HTTPS 或者基于 SSH 的通信过程无法进行监控。如果入侵通过这些加密的通信过程进行,则网络入侵防范无法有效地检测入侵并进行响应,此时主机入侵防范可以在用户终端通过对解密后的通信内容检测而发现可能的入侵并进行响应。

参考文献

1. 龚俭,吴桦,杨望. 计算机网络安全导论. 南京:东南大学出版社,2009

2. Scarfone K, Mell P. Guide to intrusion detection and prevention systems (IDPS). NIST Special Publishing. 2007

(杨望)

wangluo ruqin jiance

网络入侵检测(network intrusion detection)

网络入侵检测系统(intrusion detection system, IDS)是使网络入侵检测和响应过程自动化的一种**网络安全**设施,它可以是系统中的一个功能软件,也可以是一个独立设备。网络入侵检测是对入侵的识别,IDS 通过在计算机网络或计算机系统中的若干关键点收集信息并进行分析,试图从中发现网络或系统中是否有违反安全策略的行为和遭到攻击的迹象。通过网络入侵检测能使安全管理员发现入侵行为的存在,以便其采取适当措施来尽可能减少入侵对系统造成的损害。IDS 本身并不需要对攻击做出反击动作,而仅仅尽力感知攻击或攻击尝试的存在,因此对于 IDS 而言的响应指的是对检测结果的报告能力。

IDS 的构建基本上有两种方法:基于网络的入侵检测系统(NIDS)和基于主机的入侵检测系统(HIDS),两者的差别主要是数据来源不同。基于主机的入侵检测系统从单个主机上提取数据(如审计记录等)作为入侵分析的数据源,而基于网络的入侵检测系统从网络上提取数据(即网络报文)作为入侵分析的数据源。

入侵检测系统通常分为滥用检测(misuse detection)和异常检测(anomaly detection)两类。两者的区别在于检测的理念。如果定义什么是异常,而其余事件被视作正常,这就是滥用检测;如果定义什么是正常,那么所有不符合正常的事件都是异常事件,这就是异常检测。如果 IDS 没有发现存在的异常事件,称为漏报;将正常事件误认为异常事件,称为误报。**漏报率**和**误报率**均是评价 IDS 的重要指标。

滥用检测又称误用检测,也可称为基于知识的检测或者模式匹配检测。它的前提是假设所有的网络攻击行为和方法都具有一定的模式或特征,如果把以往发现的所有网络攻击的特征总结出来并描述在一个入侵特征库中,那么 IDS 可以将当前捕获到的网络行为特征与入侵特征库中的特征信息相比较,如果匹配,则当前行为就被认定为入侵行为。因此,滥用检测的工作原理与防病毒软件的工作原理非常相似,著名的开源系统有 Snort 和 Bro。

异常检测也称为基于行为的检测,是指根据用户行为和系统资源的使用状况判断是否存在网络入侵。异常检测技术首先假设网络攻击行为是不常见的或是异常的,可区别于所有的正常行为。如果能够为用户和系统的所有正常行为总结活动规律并建立相应的行为基准模型,那么 IDS 可以将当前捕获到的网络行为与基准模型相对比,若入侵行为偏离了正常的行为轨迹,就可以被检测出来。异常检测的实现方法有很多,主要集中在基于统计理论的检测模型和基于机器学习理论的检测模型这两个方面。

参考文献

龚俭,吴桦,杨望. 计算机网络安全导论.

2 版. 南京：东南大学出版社，2007 （龚俭）

wangluo sheji

网络设计（network design） 制定网络解决方案的一个迭代过程：包括客户需求分析，逻辑网络设计，物理网络设计，测试、优化和文档编写等 4 个阶段。旨在保证一个新的网络或服务方案能满足客户和运营者的需要。通常采用自顶向下的结构化设计方法。

客户需求分析阶段

主要任务有：①明确客户的商业目标，识别网络设计的范围和客户的网络应用，了解影响设计的约束条件；②掌握和描述现有网络的基础设施特征和网络的流量特征。前者包括网络结构、编址和命名、布线与介质、建筑结构与环境制约以及它们的健壮性等；后者包括主流量源和存储、流量类型、负载估算、流量行为和服务质量需求等；③分析客户的网络技术目标，包括网络的可扩展性、可用性、网络性能、安全性、可管理性、易用性、适应性和可购买性等，并根据约束条件进行折中选择。而网络性能包括吞吐量、正确率、效率、延迟、抖动和相应时间等。作为需求分析的结果，应分别填写相应的检查表。

逻辑网络设计阶段

主要任务有：①按经典的核心层、分布层和接入层的三层模型设计网络拓扑结构，其中应包含相应各层的备用路径和负载分担的冗余拓扑。按模块化原则设计园区（或企业）网络的主干、楼宇汇聚与接入的拓扑结构。设计园区网络接入到广域网、多宿主 Internet、虚拟专用网或服务提供商等的边界拓扑结构。设计保护网络的安全拓扑结构。②为网络组件（包括网络、子网、路由器、交换机、服务器和终端系统等）设计编址和命名的规则。前者包括网络层地址的分配原则以及层次化编址与路由选择；后者包括命名及其分布授权的原则，从名字到地址的透明映射存在静态和动态两种方法，但一般采用后者，如 DNS 域名解析系统。③选择交换和路由协议，前者包括透明桥接、多层交换、增强 STP 以及 VLAN 等协议；后者包括距离向量路由协议 RIP、BGP 等以及链路状态路由协议 OSPF、IS-IS 等。通常，客户的网络性能要求、预计的网络流量特征和交换机与路由器的能力将影响或决定这些选择。④设计网络安全策略，包括明确网络资产和风险、分析安全折中、制定安全规划、策略和流程；制定物理安全、认证、授权、审计、数据加密、包过滤、部署防火墙和入侵检测系统等的保障机制；对特别网络业务和应用实施模块化安全保护，包括对公共服务器和电子商务服务器、远程访问和虚拟专用网、网络业务和网络管理、服务器集群、用户业务以及无线网络的保护等。⑤制定网络管理策略，包括对性能、故障、配置、安全和记账等 5 类网络管理的流程设计；网络管理体系结构的选择；网络管理协议和工具的选择等。

物理网络设计阶段

主要任务有：①为所设计的逻辑网络选择具体的互连设备和互连媒质，前者包括集线器、交换机、路由器、无线接入点等设备以及这些设备内的软硬件配置；后者包括屏蔽双绞线、屏蔽铜轴电缆、屏蔽双轴电缆、非屏蔽铜缆、单模光纤和双模光纤，及其布线设计等。②为逻辑网络选择远程接入广域网的设备和通道技术，前者包括接入服务器、路由器和 VPN 集中器等；后者包括裸光纤通道、数字专线、POS、帧中继、ATM 或城域以太网（GE）等。③为客户提供网络设备、通道以及服务的参考价格。

测试、优化和文档编写阶段

主要任务有：①对网络设计进行测试，以验证是否满足客户的商业目标和技术目标以及端到端的性能指标和服务质量要求。把新网络的初始配置作为原型系统进行测试，编写包括测试目标、验收标准、测试种类和测试对象在内的详细测试计划，并按测试脚本推进测试计划的执行。通常按性能、压力和故障等进行分类测试。性能测试主要检测网络的吞吐量、延迟、抖动、响应时间和效率等指标；压力测试主要检测网络负载增加而导致的服务质量下降的情况；故障测试主要检测网络可用性和误码率，并分析其原因。也可对网络的管理能力、可用性、适应性和安全性等其他技术目标要求进行测试。②根据测试结果对网络进行结构修正和优化设计。优化技术包括使用组播技术优化网络骨干带宽的利用；通过链路层的分片和交织（LFI）、多媒体数据包报头压缩等技术减少广域网链路串行化延迟；通过调整 IPV4 报头中的优先级和服务类型、IPV6 报头中的流标签、资源预留协议（RSVP）和公共开放策略服务协议（COPS）中的 QoS 参数等优化网络性能，以满足服务质量要求等。③按相关标准和客户要求编写网络设计文档，主要包括上述各个阶段的设计成果和测试结果以及推荐的网络运行、管理和升级方案等。

网络设计是一项十分复杂的工程任务。当一次

设计难以成功时,可能会从头开始进入新一轮的网络设计,直至满足客户的需求。

参考文献

Priscilla Oppenheimer. 自顶向下网络设计. 2版. 胡捷,译. 北京:人民邮电出版社, 2005

(李芝棠)

wangluo shipeiqi

网络适配器(network adaptor)　将计算机、存储设备、**终端设备**、移动设备等连接到**互联网**,并进行通信的接口装置。又称**网卡**(network interface card,NIC)或网络接口控制器。是计算机、存储设备、智能终端、移动设备等设备与网络连接的桥梁,具有十分重要的作用。

网络适配器结构

由硬件和固件程序组成。硬件包括PCB电路板、收发器、数据链路层控制器、网络隔离变压器、晶体振荡器、总线插槽接口(金手指)、BOOTROM、EEPROM、RJ45接口、信号指示灯、固定片以及一些二极管、电阻、电容等。固件程序是内嵌在**只读存储器**中的一段程序,主要实现逻辑链路控制和媒体访问控制功能,并记录了一个唯一的硬件地址,即MAC地址。每一个网络适配器都有一个世界上独一无二的MAC地址,该地址是一个48位的串行号,用于标识网络中设备的身份,由IEEE(美国电气电子工程师协会)负责为每个网络适配器分配一个唯一的MAC地址。

网络适配器的基本功能

(1)将计算机中的数据封装成帧,并通过网线(有线或无线)将数据发送到网络上。

(2)接收网络上传送过来的帧,并将帧重新组合成数据,传递到网络适配器所在的计算机。因为网络传输介质一般只能传输串行的模拟信号,因此网络适配器不仅要实现与网络传输介质之间的物理连接和光/电信号匹配,还要完成数据帧的封装与拆封、串并转换、数据帧的发送与接收、介质访问控制、数据编码与解码、数据缓存等功能。

网络适配器工作原理

(1)网络适配器工作在OSI(开放系统互连)7层模型的最后两层,即物理层和数据链路层。物理层定义了数据传送与接收所需要的光与电信号、线路状态、时钟基准、数据编码和电路等,并向数据链路层提供标准接口;数据链路层则定义了寻址机构、数据帧的封装和解封、数据差错校验、传送控制等,并向网络层提供标准接口。网络适配器通过物理层和数据链路层,将用户要传递的数据转换为网络上其他设备能够识别的格式,通过网络传输介质,完成设备之间的通信。

(2)网络适配器的数据发送过程　数据链路层接收到网络层传送来的数据包后,将之拆分并重新打包成最大1518字节、最小64字节的帧,传送给物理层;物理层接收到数据链路层传送过来的数据(对物理层来说,没有帧的概念,都是数据而不管是地址、数据还是循环冗余检验码)后,进行差错控制(每4比特就增加1比特的检错码),然后把并行数据转化为串行数据,再按照物理层的编码规则对数据进行编码,最后转变为模拟信号,以差分方式传送出去。接收过程则相反。由于网络上连接了大量不同的网络设备,设备之间的电网环境、电势水平不同以及周围环境的电磁感应和静电,很容易造成芯片的损坏,为此引入了网络隔离变压器(transformer),把物理层传送出来的差分信号用差模耦合的线圈耦合滤波以增强信号,并通过电磁场的转换耦合到连接网线的另外一端,这样不但使网线和物理层之间没有物理上的连接,而且转换传递了信号,隔断了信号中的直流分量,可以保证在不同基准电平的设备中传送数据。

(3)帧数据传输　网络适配器是以帧为基本单位进行**数据传输**的。帧是OSI模型中数据链路层的数据组成单位,由首部和负载组成,是一个基本传输单元。在帧中不仅包含有数据信息,而且还包含有数据的发送地、接收地信息以及数据的校验信息。例如,**以太网**帧包括目标MAC地址、源MAC地址和数据包协议类型、循环冗余检验码等。

网络适配器分类

根据传输介质和使用场景的不同,网络适配器可分为无线网络适配器、有线网络适配器、网络存储系统用网络适配器、高性能计算机用网络适配器等。

(1)无线网络适配器　通过无线信号进行连接。无线网络适配器根据接口不同,可分为PCMCIA无线网络适配器、PCI无线网络适配器、MiniPCI无线网络适配器、USB无线网络适配器、CF/SD无线网络适配器等几类。

(2)有线网络适配器　通过有线(如双绞线、光纤等)连接,可分为RJ-45接口、光纤接口、AUI接口和BNC接口等几种。其中,RJ-45接口广泛应用于以太网,光纤接口则多用于服务器,而后两种现今已经较少见。

（3）网络存储系统用网络适配器　主要完成大容量网络存储系统的数据的存储，实现分布式网络存储系统或高速计算机与高速存储设备的互连。因此需选用高性能网络适配器，以实现高可靠、高速率和低延迟的数据传输。其网络适配器可分为FC（fiber channel）光纤通道网络适配器、InfiniBand网络适配器等。

（4）高性能计算机用网络适配器　针对面向科学计算、事务处理、数据库应用、网络等应用的高性能计算机，选择具有强大外部数据吞吐能力的高端网卡，以解决I/O墙的问题，通常配备多千兆以太网网络适配器、10千兆以太网、InfiniBand网络适配器等。（薛一波）

wangluo suidao

网络隧道（tunnel）　在网络中用一种网络协议封装另一种不同协议的传输技术，其中封装其他协议的封装协议起着承载的作用，而被封装的协议是封装协议的载荷。通过网络隧道，某种载荷能够经过一种不兼容的承载网络进行传输，或实现经公共网络的安全通路。

封装协议与被封装的协议之间并没有层次的限制，即高层协议也可以封装较低层的协议。不过，根据传统的分层协议如OSI模型或TCP/IP模型所实现的协议封装，如FTP在TCP之上，TCP在IP之上，IP在以太网协议之上，这些都不认为是网络隧道，因为这是相应网络体系结构所规定的正常承载方式。

TCP/IP网络中两个IPv6网络经IPv4网络互联是应用网络隧道技术的一个例子。IPv6是不同于IPv4的一种新型互联网协议，尽管它们在TCP/IP网络中位于相同层次，但两个协议互不兼容。当IPv6分组经网关进入IPv4网络时，IPv6（载荷协议）分组被封装为IPv4（交付协议）的载荷，采用协议编码41来标识，通过IPv4网络来传输。尽管IPv6是由RFC 2460定义的协议，但此时它只能使用由RFC 791定义的IPv4协议来传输。此时，交付协议和载荷协议及其地址都是不兼容的。当该分组到达IPv4网络与IPv6网络的边界时，网关将从IPv4载荷中恢复出IPv6分组，交由IPv6网络传输。

二层隧道协议（Layer 2 Tunneling Protocol，L2TP）是一种当前得到广泛支持的隧道标准。当IP分组承载在L2TP中时，通过L2TP报文交换实现了L2TP连接的维护以及PPP数据的传送功能，这些报文再通过UDP的1701端口承载于IP之上。L2TP报文可以分为两种类型，一种是控制报文，另一种是数据报文。控制报文用于建立与维护隧道连接和会话连接，数据报文则用于承载用户的PPP会话数据分组。

隧道协议可以使用数据加密技术来传送不安全的载荷协议经过公共网络如因特网，从而提供了VPN功能。例如，IPsec采用端到端传送方式，借助于可信的安全网关以隧道方式保证IP分组以所需要的某种安全方式进行传输。

参考文献

1. Kurose J F, Ross K W. Computer networking. 5th ed. Boston, MA: Addison Wesley, 2010

2. Tanenbaum A S. Computer Networks. 4th ed. Englewood Cliffs, NJ: Prentice Hall, 2003（陈鸣）

wangluo tixi jiegou

网络体系结构（network architecture）　计算机之间相互通信的层次、各层次中的协议和层次之间接口的集合。**计算机网络**的设计是按高度结构化方式进行的。为了减少网络设计的复杂性，增加网络系统的开放性和互操作性，几乎所有的计算机网络都按分层方式组织和设计协议。层是指网络体系结构中功能划分明确的一个部分。

不同类型的计算机网络具有不同的体系结构，其层的数量、各层的定义和功能以及各层之间的接口都不一样。每层都要为相邻的上层提供特定的服务，而如何实现协议和服务的具体细节都对相邻层屏蔽。这样，网络体系结构就能做到与具体实现无关，哪怕连接到网络中的主机的型号和性能各不相同，只要它们共同遵守相同的协议就可以实现互相通信和互相操作，从而构成开放的网络系统。

网络中一台计算机和另一台计算机的通信必须在同一层进行。通信所用的规则和顺序以及所传递消息的格式构成该层的协议（参见**网络协议**）。计算机网络中的数据传送方式不是从发送方的第 n 层直接传送到接收方的第 n 层，而是每层都把数据和控制信息传递给它的下一层，直到物理传输介质。接收时，则是每层从它的下一层接收相应的数据单元，并去掉与本层有关的控制信息之后把剩下的数据交给它的上一层。

各个相邻层之间都有相应的接口，该接口定义了下层向上层提供的各种服务和使用这些服务的操作和响应。为了保证这些操作和响应的功能完整

性，它们被设计成在执行过程中不允许中断或不允许并发执行的原语。网络设计者在设计网络体系结构时，首先必须要定义每层所要完成的功能集合，然后定义上下层之间的接口，最后设计为完成所需要功能与服务的协议。清晰的接口和具有明确含义的功能集合不仅使协议的设计和实现变得容易，而且使得在相同层中用一种协议实现代码代替另一种协议实现代码成为可能，因为只要这两种实现代码能为上层提供相同的服务即可。

层的划分必须适当，层次太多会造成系统处理时间增加和报文头长度增加，导致系统开销增加，这在那些要求高速传输的网络中是不允许的。但是，层次太少又会造成每层的功能不明确，相邻层间接口不易确定，从而使得协议的可靠性降低。大部分网络体系结构的层次为4～7层。

首先提出计算机网络体系结构概念的是美国国际商用机器（IBM）公司。IBM公司于1974年提出了系统网络体系结构SNA。1978年国际标准化组织ISO提出了**开放系统互连（OSI）参考（基准）模型**，并陆续推出了有关协议的国际标准，从而确立了OSI网络体系结构。这为不同制造商的计算机与不同的计算机网络之间的互连提供了依据，因为只要这些计算机和网络遵守OSI体系结构和相关协议，它们就能够互相通信，互相操作。

OSI网络体系结构在很大程度上模仿了SNA网络体系结构。OSI网络体系结构被分为七层，自底向上分别是：①物理层：负责在计算机之间传递原始比特流。②数据链路层：把原始比特流组成帧，检测传输错误，提供无差错传输。③网络层：控制通信子网，选择合适的路由，将分组从源主机转发到目的主机。④传输层：提供端到端的数据传送服务。⑤会话层：为用户提供会话控制服务，例如令牌管理和同步控制。⑥表示层：为用户提供数据转换和表示服务。⑦应用层：包含用户使用的各种网络应用。低层为相邻高层提供服务。报文从上而下进行发送，从下而上进行接收。每层协议都把相邻高层送来的报文作为数据信息，加上该层的控制报头之后交给相邻低层。

最初由美国国防部支持开发的ARPANET，现已演变为连接全球的互联网，该网络使用TCP/IP协议。互联网体系结构分为五层，分别是物理层、数据链路层、网络层、传输层和应用层。

计算机网络体系结构在不断发展。当前，以TCP/IP协议为核心的层次划分的网络体系结构是计算机网络体系结构的主流。近些年，关于未来互联网体系结构的研究方兴未艾。

参考文献

1. 张尧学，等. 计算机网络与Internet教程. 2版. 北京：清华大学出版社，2006

2. Tanenbaum A S, Wetherall D J. 计算机网络. 5版. 严伟，潘爱民，译. 北京：清华大学出版社，2012

（张尧学　徐明伟）

wangluo xieyi

网络协议（network protocol）　**计算机网络**和分布式系统中互相通信的对等实体间交换信息时所必须遵守的规则的集合。这里，对等实体是指在计算机**网络体系结构**中处于相同层次的通信进程。协议具有和计算机语言几乎完全相同的定义，即协议为传输的报文定义严格的格式（语法）和传输顺序（文法），而且协议还定义所传输报文的词汇和这些词汇所表示的意义（语义）。

（1）协议的通信环境　网络协议可被抽象成一个层次模型。在层次模型中，低层协议为相邻的高层协议提供服务，高层协议调用相邻的低层协议提供的服务完成对等实体之间的信息交换功能。两个协议对等实体通过其低层协议构成一个通道。从而，对于一个n（$n \geqslant 2$）层协议来说，其低层协议构成（$n-1$）层通道。另外，n层协议还为本层协议的用户（简称n层用户）提供服务。用户要求、通道性质以及n层协议运行时的操作系统和硬件条件等构成了n层协议的通信环境，如图1所示。

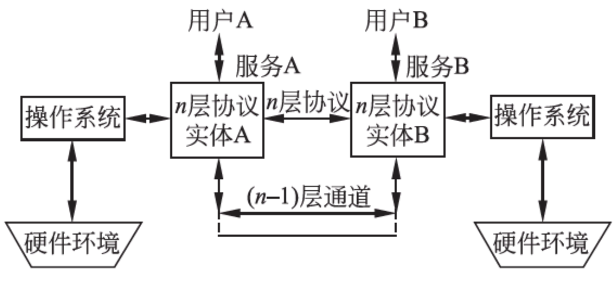


图1　n层协议的通信环境

（2）服务（services）　第n层协议所提供的服务是第n层协议的外部行为体现，是第n层协议向它的高层用户提供的数据传输方式和相关的处理方式。数据传输服务包括面向连接的服务和无连接的服务。除了数据传输方式的服务之外，还有网络传输质量方面的服务，例如时延大小、抖动程度等。服

务也是分层的。服务规范被用来定义和描述第 n 层协议提供的服务,描述服务使用者和提供者之间交换作用的规则。面向连接的服务在计算机开始通信之前,必须在通信网络之间建立一条端到端的通信路径,然后开始数据通信;待数据通信结束后,再终止这个连接。面向连接的服务可分为三个部分:建立连接、数据传输和断开连接。面向连接的服务又分为永久性连接服务和非永久性连接服务。非永久性连接服务在数据流传输之前,都要先进行连接,待连接成功后再进行通信。永久性连接则是在第一次数据流传输前进行连接,待该连接成功后,把相应的连接路径存入计算机,以后的通信中不再进行连接。无连接的服务中,无论何时,计算机都可以向网络发送数据包,而无须事先建立连接。无连接的服务方式传输的每个数据包中必须包含目的地址,以便寻找合适的路由。与面向连接的服务相比,无连接的服务占用网络资源少,处理开销小,发送消息快,但可靠性较低。

(3) 词汇表 词汇表定义 n 层协议中所使用的消息以及它们的意义。例如,“ack”可被定义成接收方正确接收到数据单元后的应答,“nak”则可被定义成未正确接收到数据包或传输出错(未到达接收方)时的应答。

(4) 消息的编码格式 消息的编码格式是协议的语法定义,它包括数据长度、控制信息长度以及每个域的定义等。**网络数据单元**(network data unit)是网络中一次传输的数据单位,包括协议数据单元(PDU)和服务数据单元(SDU)。SDU 是相邻层实体间传送信息的数据单元,PDU 是对等实体间传送信息的数据单元,它是 SDU 加上同层协议控制信息(PCI)所形成的数据单元。SDU 即可以封装成一个 PDU 传输,也可以被分成几段后,每段加上一个 PCI 组成新的 PDU 传输。$n-1$,n,$n+1$ 层数据单元 SDU 与 PDU 的关系如图 2 所示。($n+1$)PDU 是借助(n)SDU 传到 n 层的,此时(n)SDU 就相当于 n 层的用户数据,对它加上(n)PCI 后便构成了(n)PDU。这里,($n+1$)PDU 似乎等同于(n)SDU,实际上,两者往往是不同的。有时,发送方实体需要将数个($n+1$)PDU 拼接成一个(n)SDU,而在接收方对等实体需进行一个(n)SDU 分割成数个($n+1$)PDU 的操作。

(5) 时序、规则和过程 时序、规则和过程是网络协议中最复杂、最关键的部分,它们规定用什么样的方法和算法去完成服务规范所定义的协议功能。

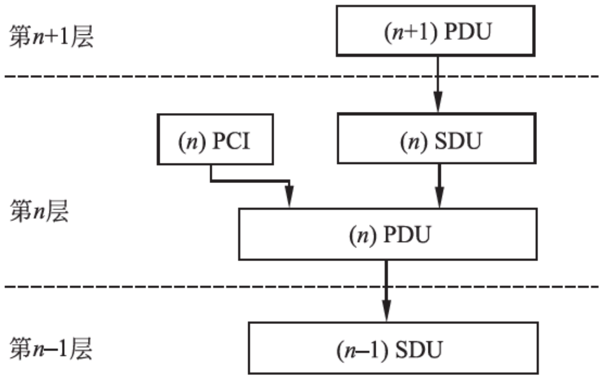


图 2 SDU 与 PDU 之间的关系

协议的功能除了包括连接管理、通信方式管理、协议数据单元的发送和接收以及装配和拆卸等之外,还包括数据单元的编码和解码、分解和组合以及流量控制、拥塞控制、发送顺序控制、发送速度控制和出错处理等。协议的规则和过程被用来完成这些工作。

网络协议规范(network protocol specification)是由协议功能和服务组成的关于协议的通信顺序、逻辑、格式和行为动作的描述。网络协议规范由相应的国际标准化组织颁布。这些组织包括国际电信联盟(ITU)、国际标准化组织(ISO)以及 Internet 工程任务组(IETF)等,这些组织接受和审查由科学家和研究小组提出的各种协议标准提案,提案经过会议讨论和表决通过后成为相应的国际标准规范。

参考文献

1. 张尧学,等. 计算机网络与 Internet 教程. 2 版. 北京:清华大学出版社,2006

2. Tanenbaum A S, Wetherall D J. 计算机网络. 5 版. 严伟,潘爱民,译. 北京:清华大学出版社,2012

(张尧学 徐明伟)

wangluo xieyi gongcheng

网络协议工程(network protocol engineering) 一门研究如何设计和构造协议规范以及如何把所设计和构造的协议规范快速、准确、低成本地转化为可执行代码的科学。它为协议的设计和开发提供一套综合的、规范的、自动的方法。协议工程包含两方面内容,一方面是协议规范(包括语法、文法和语义)的制定,另一方面是高效率、低成本的自动

或半自动的协议软件开发方法的研究。网络协议开发的过程与步骤如图 1 所示。

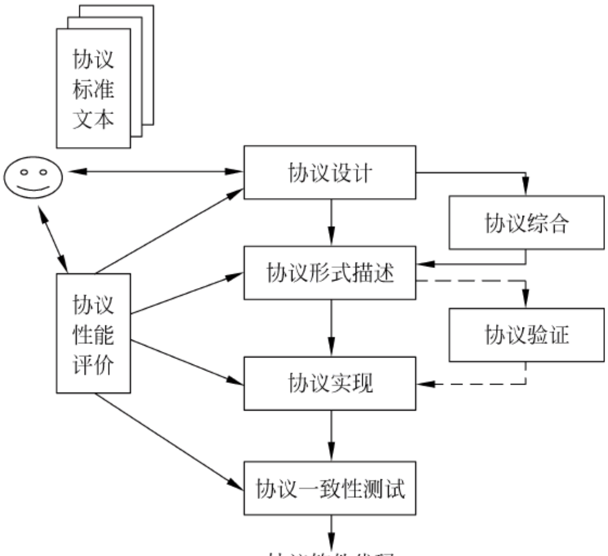


图 1　协议开发过程与步骤

协议规范与标准的制定大多由国际标准化组织负责,例如国际电信联盟(ITU)、国际标准化组织(ISO)以及美国电气与电子工程师学会(IEEE)等。而与 Internet 有关的各种协议规范与标准的制定则由 Internet 协会下属的互联网工程任务组(IETF)负责。IETF 根据 Internet 发展的技术趋势,把 Internet 相关技术划分为 10 多个领域,设置了 100 多个工作小组,这些工作小组在 Internet 上靠电子邮件互相交换信息并进行联系,对共同感兴趣的协议进行研究,然后提出相应的请求评论(RFC)提案草案或标准。协议工程的另一个内容是软件开发过程的工程化,主要研究协议的开发方法。由于协议开发仍是软件的开发,因此,软件工程的方法也可用来开发协议软件。协议软件与一般软件的主要区别表现在:

(1) 协议软件运行在分布式网络环境下,比较复杂,规模也较大。

(2) 协议软件具有开放性,对标准化要求高。

(3) 由于是分布式网络环境下运行的软件,一般具有较多的潜在错误,且难以发现和排除。

(4) 协议软件的实现依赖于网络所提供的硬、软件平台和环境,软件移植性较差。

(5) 安全性要求高。

协议开发必须包括以下几个方面:

(1) 协议设计　协议设计包括概要设计、结构和模块设计、实现环境分析和详细设计、验证与测试过程设计等。协议设计的基础是协议标准规范。它把一个用自然语言或标准形式描述语言表示的协议按照协议实现的要求从需求分析开始,逐步设计成一个符合实现平台需要的详细描述。

(2) 协议综合　协议综合从协议规范的文本出发,通过相应的综合规则和算法,抽取出协议实现所必需的功能、交互顺序和各模块所提供的服务,得到协议的形式描述,使其不包含逻辑错误,并且具有活性与安全性等好的性质。

(3) 协议验证　通过数学方法和工具对所设计或描述的协议进行检验,证实所设计和描述的形式描述中不存在逻辑错误,即死锁、不完全性、不稳定性、活锁和信道溢出等,从而确认协议的形式描述具有安全性和活性。

使用协议综合技术设计后得到的协议形式描述不需要再进行协议验证工作,因为协议综合时所用的规则与方法已经保证了所得到的协议形式描述具备了安全性和活性。但是,由于协议综合技术很难应用于大型协议的设计过程,因而,人们仍然采用先按照协议标准文本进行形式描述,然后再将协议形式描述抽象为有限状态自动机(FSM)、Petri 网(描述系统各元素的异步并发操作的工具模型)或抽象数据类型等数学模型进行验证。在验证过程中如果发现协议的形式描述中存在相应的错误,则对其进行修改后再重新验证,这一过程一直重复到验证结果中不再发现逻辑错误为止。

(4) 协议实现　协议实现把经过验证或综合设计的协议规范变换成计算机可执行代码,这是一个从形式描述到软件代码的逐步精化过程。协议实现要求首先把协议标准文本或规范抽象为面向实现的协议规范,面向实现的协议规范包括协议的应用范围、协议各层之间的服务定义、接口及功能子集的定义、与实现有关的各种参数规定、服务质量控制信息以及一致性测试条件等。协议实现方法有计算机工具辅助的半自动实现方法和完全手工编程的实现方法两种。计算机工具辅助的半自动实现方法需要有相应的形式描述语言编译器,生成的软件代码相对较长。手工编程实现的软件代码比较精练,但可能含有较多的错误。

(5) 一致性测试(参见**网络协议一致性测试**)

参考文献

1. 张尧学,等. 计算机网络与 Internet 教程. 2 版. 北京:清华大学出版社,2006

2. Tanenbaum A S, Wetherall D J. 计算机网络. 5 版. 严伟,潘爱民,译. 北京:清华大学出版社, 2012

(张尧学 徐明伟)

wangluo xieyi xingshi miaoshu jishu

网络协议形式描述技术(network protocol formal description technology) 用形式描述语言描述协议实体间信息交互规则、格式和相关定义的技术。协议形式描述主要描述三大部分,即协议实体所提供的服务、协议实体的交互行为和该协议实体所提供的服务如何使用,也就是协议实体的用户界面。

形式描述技术的基础是基于数学的形式描述语言,例如 SDL、Estelle、Lotos 和 Z 语言等。使用这些形式描述语言对协议文本和规范进行描述,可以减少或消除由自然语言描述所带来的歧义性与不确定性。协议文本和规范中的歧义性与不确定性容易导致协议软件代码中包含无法预料的逻辑错误,并使得不同的实现人员编写出的软件代码具有不同功能,从而损害网络的互通性和互操作性。

除了上述形式描述语言之外,ASN. 1 是一种经常被用来描述应用层协议的形式描述语言。例如, **Internet** 的**简单网络管理协议**(SNMP)以及**管理信息库**(MIB)就是用 ASN. 1 描述的。形式描述语言树表结合表示法(TTCN)则被用于描述一致性测试中的协议测试集。

参考文献

1. 张尧学,等. 计算机网络与 Internet 教程. 2 版. 北京:清华大学出版社, 2006

2. Tanenbaum AS, Wetherall DJ. 计算机网络. 5 版. 严伟,潘爱民,译. 北京:清华大学出版社, 2012

(张尧学 徐明伟)

wangluo xieyi yizhixing ceshi

网络协议一致性测试(network protocol conformation testing) 验证协议实现的源程序执行时所提供的功能是否与协议标准文本所规定的功能相一致的测试。

进行一致性测试的目的是为了满足网络互联的要求,即不同的人员根据同一标准开发出来的协议软件可以通过网络进行互连和互操作。一致性测试并不检验协议标准文本中是否存在逻辑错误,它只负责检查协议的各实现版本是否能够完成协议标准所要求的功能。

进行一致性测试需要有被测实体,也就是被测协议软件、测试环境以及要求被测实体所完成功能的协议测试集。这些协议测试集被当作用户要求输入,由被测实体在测试环境下运行和测试,然后对测试结果进行分析,如图 1 所示。

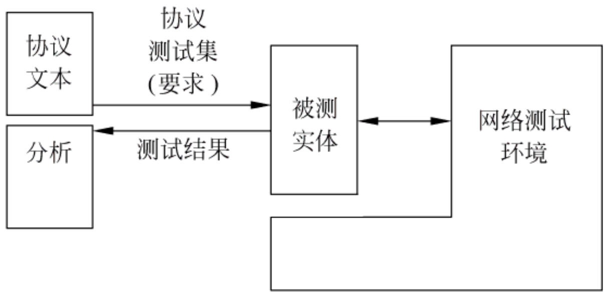


图 1 一致性测试概要

为了确认被测实体所实现的功能是否与协议标准所规定的功能相一致,必须定义和明确协议实体所应该具有的要求和功能。这些要求和功能被称为一致性要求。一致性要求包括如下四点:

(1) 强制性要求 在任何情况下都要进行检查的要求;

(2) 条件性要求 在标准规定的具体条件下必须满足的要求;

(3) 任选要求 根据具体实现不同可进行选择的要求;

(4) 禁止要求 禁止进行检查的要求。

这些要求又可分为静态一致性要求和动态一致性要求。

静态一致性要求是为了方便网络互联而要求被测协议实体所应具有的最小能力。动态一致性要求定义被测协议实现所应具有的所有可能的行为。

一个被测实体既满足静态一致性要求又满足动态一致性要求的某个指定部分的话,则认为该协议实现与协议标准规定的功能在指定范围内是一致的。

一般分四个方面进行协议的静态测试和动态测试,它们是:

(1) 基本互连测试 确定被测协议实体是否具有初步互连能力;

(2) 能力测试 对被测协议实体进行静态一致性测试,并与静态一致性要求相比较;

(3) 特性测试 对被测协议实体进行动态一致性测试,并将所得结果与动态一致性要求相比较;

（4）判定测试　对某个具体部分进行的小范围测试，以进一步确认在特性测试中不能确定一致性而又可能满足某些一致性特定要求的部分。

为进行上述测试，ISO 定义了 TTCN 形式描述语言，描述被测协议实体的静态一致性与动态一致性要求。许多测试方法和工具被开发出来用于协议的一致性测试，根据这些方法，测试人员首先从 TTCN 描述的一致性要求中抽象测试集和选择测试方法（因不可能对所有的一致性要求进行完整的覆盖性测试）。

测试集的选择要有代表性，否则会漏掉协议实现一致性要求中的某些重要性质。

测试方法的选择除了测试算法之外，还包括测试环境的选择。ISO 定义了四种基本的测试方法，即局部测试法、分布式测试法、协调测试法和远程测试法。与此对应的测试环境有三种：局部环境、远程环境和分布式环境。这些方法分别将被测协议实体放在不同的网络环境中进行一致性测试和观察。

在局部测试法中，被测协议实体的上下层协议实体和对等通信协议实体都处在同一主机中。被测实体与上层协议实体以及对等协议实体的交互界面可以由测试人员直接控制和观察，如图 2 所示。

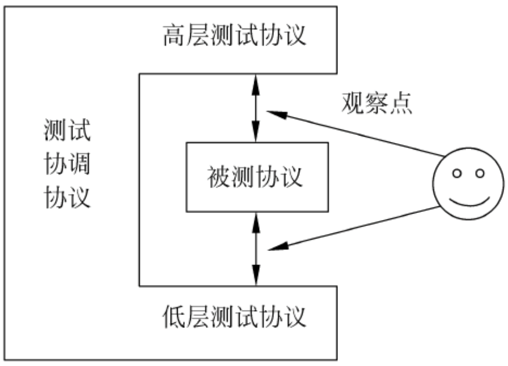


图 2　局部测试法

分布式测试法使被测协议可以置身于一个如图 3 所示的网络环境中，测试人员通过执行所选择的测试集，在低层测试协议端控制和观察被测协议实体调用低层协议的服务和行为的一致性，而在高层测试协议端控制和观察被测协议实体为高层提供服务和行为的一致性。所谓的低层测试协议实际上是被测协议的对等实体，即和被测协议处于同一层次的协议。

协调测试法是分布式测试法的增强形式。协调测试法在高层测试协议端对被测协议实体的行为进行控制和观察，这种控制和观察改在低层测试协议端进行。

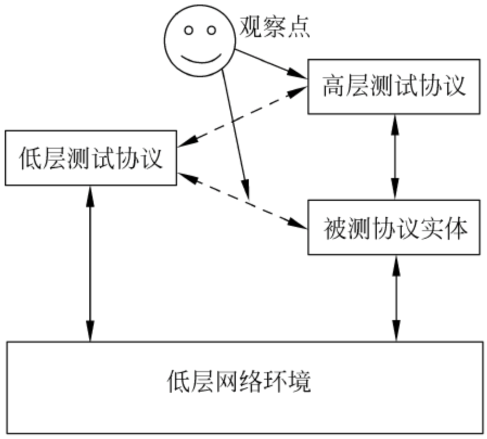


图 3　分布式测试法

远程测试法更进一步放弃了在低层测试协议端对被测协议与高层测试协议的交互行为的控制和观察。

参考文献

张尧学，等. 计算机网络与 Internet 教程. 2 版. 北京：清华大学出版社，2006　（张尧学　徐明伟）

wangluo yidong

网络移动（network mobility）　网络移动性支持是指网络整体的移动性管理，移动网络作为独立的单元，能够改变它与互联网的连接点并因此改变它在互联网拓扑中的可达性。移动网络包含一个或多个与互联网相连接的网关——*移动路由器*（MR），它的移动性只能通过移动路由器来管理，在移动网络中，MR 后方的移动网络节点（MNN）既可以是固定的（本地固定节点 LFN），也可以是移动的（本地移动节点 LMN 或者来访移动节点 VMN）。在一般情况下，移动网络的内部结构是相对静止的（即没有动态变化的拓扑），但是这点并不总是成立。

移动网络的实例如下所示：

个人局域网络（PAN）　一个拥有蜂窝接口和蓝牙接口的移动电话再加上一个可以使用蓝牙的 PDA 就可以构建一个简单的移动网络实例，其中移动电话作为移动路由器而 PDA 用于浏览网页或者运行个人网络服务器。

汽车内的传感器和计算机网络　为了提高安全性和降低驾驶难度，汽车安装了越来越多的处理单元

与传感器，传感器收集车辆的性能、状况等信息，并与处理单元之间相互通信，对信息进行处理，同时还可以在行驶过程中通过车内的无线路由器与周围的车辆或路边的公共网络设备相互通信，例如ITS（智能交通系统）应用。

公共交通工具（公共汽车、火车、出租车、飞机）内的接入网络 在公共交通工具内部配备有能够连接到互联网的无线路由器，从而为拥有IP设备的乘客提供互联网接入（笔记本电脑、照相机、移动电话）。

通过MR连接到互联网的ad-hoc网络 举例说明，火车上的一些学生可以既在他们自己之间建立一个ad-hoc网络，也可以通过将火车连接到互联网上的MR来获得互联网接入服务。

网络移动性支持是移动网络节点和所对应的通信点（CN）在移动路由器改变网络接入点的过程中保持会话持续性的机制，其目的是保证异构网络之间的漫游以及无缝移动，为移动网络提供相应的移动性管理，使移动网络与互联网的连接保持在最佳状态。为了解决这一问题，将其划分为两个子问题，分别为网络移动性基础支持与网络移动性扩展支持。

网络移动性基础支持是通过利用MR与HA之间的双向通道来维持会话的。在网络移动性基础支持中，移动路由器在家乡代理处注册时获得一个唯一的家乡地址，家乡地址的前缀代表着它所对应的家乡代理，当移动路由器远离家乡链路并和新的接入路由器相连接时，它将通过所访问的链路获得一个转交地址，并立刻向它的家乡代理发送一个绑定更新报文，当家乡代理收到该绑定更新报文时创建一个从该移动路由器的家乡地址到转交地址的映射。由于移动路由器还需要为移动网络内部的节点提供数据转发服务，因此家乡代理通过在绑定更新中设定一个标记（R）来实现为移动网络内部节点转发数据。之后家乡代理向移动路由器发送绑定确认报文，一旦绑定流程结束，家乡代理与移动路由器之间就建立起了一个双向通道。当对应通信节点发送数据包给移动网络中的节点时，数据包首先被发送到该移动网络的家乡代理，家乡代理对数据包进行封装并转发给移动路由器，移动路由器收到封装之后的数据包，在分析确认该数据包外层头部的发送地址是它的家乡代理，目的地址是移动网络的内部节点之后将其拆封，并最终转发给移动网络内部的节点。

网络移动性扩展支持则提供了性能优化，包括变化的MNN和CN之间的路由优化。

近些年来，随着互联网的规模不断扩大以及移动通信的飞速发展，越来越多的移动用户希望能够以更加灵活、方便的方式接入网络，这就对互联网对节点移动性的支持提出了很大的挑战。为此IETF成立了NEMO（Network Mobility）工作组以解决关于移动网络的问题，并且致力于提出一个基于移动IP隧道原理的移动网络协议。

参考文献

1. Johnson D, Perkins C, Arkko J. Mobility support in IPv6. RFC 3775, June 2004
2. Devarapalli V, Wakikawa R, Petrescu A, et al. Network mobility (NEMO) basic support protocol. RFC 3963, January 2005
3. Ernst T, Lach H. Network mobility support terminology. RFC 4885, July 2007
4. Ernst T. Network mobility support goals and requirements. RFC 4886, July 2007

（崔勇）

wangluo yingyong

网络应用（network application）

指利用计算机网络所提供的互联能力、传输能力和协作能力，来满足特定需求，运行于网络环境的各种应用软件。网络应用与单机应用的区别在于，它的应用程序和应用数据至少有一项是来自于网络。网络应用可以使用局域网、广域网、互联网等多种类型的网络，其中大多数网络应用是基于因特网（最大的一种互联网）的。网络应用和网络技术互相影响，相互促进，随着网络技术的发展，网络互联的广度和深度都在不断增强，为网络应用提供了更好的支撑；网络应用的普及和创新又为网络提出了功能和性能的需求，促进了网络技术的发展。

网络应用模式

在网络应用快速发展过程中，新的网络应用模式也在不断出现，多种模式各具特点，适用于不同类型的网络应用和网络环境。早期采用专用服务器模式，一台服务器安装网络操作系统，其他工作站为哑终端；然后发展出C/S（client/server）模式，它是一种集中管理与开放式、协作式处理并存的网络服务模式；为了提高可扩展性、便利性和跨平台能力，采用了瘦客户端，从而演变出B/S（browse/server）模式；对等（peer-to-peer）服务模式的出现解决了服务器的资源瓶颈和单点失效问题，充分利用网络边缘的闲置资源；目前流行的云计算模式可以提高资源共享的

广度和资源使用的效率，它是一种按使用量付费的模式，这种模式提供可靠的、便捷的、按需的网络服务。

网络应用分类

网络应用有多种分类方式，其中根据使用者的角色不同可以分为企业应用和个人应用。

在企业应用领域，资源共享应用可以让网络内的任何人远程地访问位于不同地点的所有程序、设备和数据，从而消除了地理位置的约束，例如公司内部的办公自动化系统（OA）、企业资源计划系统（ERP）、客户关系管理系统（CRM）等。通信应用可以为员工提供功能强大的通信媒介，实现员工之间的信息交流，例如 E-mail、IP 电话、白板、桌面共享和网络视频会议系统等。此外，企业可以通过电子商务系统与客户和供应商打交道，以减少库存，提高工作效率。

在个人应用领域，网络应用更加丰富多彩。远程信息访问应用可以帮助用户方便、快捷地获得各种远程的多媒体信息，包括新闻、学习信息、娱乐信息等，例如访问新浪网站、Internet 广播电台、优酷视频网站、IP 电视（IPTV）、远程学习（telelearning）等。即时消息（instant messaging）网络应用可以实现人与人的远程通信，通信方式可以是实时的或者异步的，通信的内容可以是任何的媒体类型，例如 QQ、Microsoft Service Network（MSN）、Skype 等；除了两个人实时交流外，这些软件还提供了群聊功能，实现多人参与的消息服务；推特（Twitter）更是允许往自己的社交圈子或者其他愿意接受的观众发送文字信息；随着人与人交流的发展，出现了社交网络应用，如 Facebook、微博等，可以允许在朋友间共享个人档案和生活经历。当前，电子商务应用已经在改变人们的购物方式和习惯，顾客可以足不出户就购买到需要的商品，能够快速地搜索到需要的商品、便捷地进行不同店铺的售价比较，以提高购物的效率和体验，例如亚马逊（Amazon）、京东商城、淘宝网等。当金融业务进入电子商务后，人们可以方便地在家中管理账户、支付账单、进行投资，例如支付宝、余额宝、快捷支付等。虚拟现实网络应用可以让大量用户一起体验虚拟世界，比如谷歌地球、网络游戏、虚拟博物馆等。

另外，普适计算类网络应用正在逐步融入人们的日常生活中，通过将消费电子产品联网，可以实现智能家居系统。**物联网**进一步通过信息传感设备按照约定的协议，把任何物品与互联网连接起来，进行信息交换和通信，以实现智能化识别、定位、跟踪和管理，能够提供很多新型的网络应用，例如，生态环境、智能交通、智能电网、精准农业、物流管理、安防监控等。

移动网络应用

3G/4G 移动通信网络和 Wi-Fi 热点的普及，提供了高速的无线 Internet 接入，促进了移动互联网的快速发展。用户可以利用笔记本、平板电脑、智能手机等终端来使用移动互联网上的各种网络应用，这些应用的不断普及又促进了终端设备的发展，出现了新型的可穿戴式计算机（wearable computer），包括智能手表、智能眼镜、传感器式腕带等。得益于移动的便利性，出租车呼叫应用、手机版网站、电子邮件等应用都得到了快速发展。通过与全球定位系统（GPS）的结合，产生了许多依赖位置信息的新应用，如地图导航、位置关联信息搜索、本地天气情况等服务。与社交网络相结合，更加便利的移动社交应用逐渐成了主流，如微信、whatsapp 等。与电子商务相结合，移动商务应用（m-commerce）使得用户可以随时随地管理自己的账户，并能够通过移动电话完成小额支付。

网络应用相关理论

网络技术和网络应用在不断发展，人们对网络的认识也逐步深入，在对网络应用不断实践和探索中，总结出了很多有价值的网络定律，例如，以太网的创始人 Bob Metcalfe 提出了 Metcalfe 定律，认为网络的价值正比于用户数量的平方。在很多类型的网络应用中，用户的数量逐渐成为衡量一种网络应用价值和竞争力的主要度量，例如 Facebook 上市时的估值主要就集中在用户信息上，腾讯公司利用其庞大的用户基础可以在各个应用领域迅速占领市场。为了能够提高用户数量，有些网络应用改变了盈利模式，采用免费甚至补贴激励的方式来吸引用户，如当前的各种打车软件。用于描述人们之间联系的小世界理论，又叫六度分隔理论，假设世界上所有互不相识的人只需要很少中间人就能建立起联系，理论指出：最多通过五个中间人你就能够认识任何一个陌生人。这是社交网络的理论基础，很多社交软件利用好友关系可以迅速发展用户，进行一些广告和信息的快速传播。

参考文献

1. Tanenbaum A S, Wetherall D J. 计算机网络. 5 版. 严伟，潘爱民，译. 北京：清华大学出版社，2012

2. 谢希仁. 计算机网络. 6 版. 北京：电子工业

出版社，2013 （张海旸 马华东）

wangluo yunxing huanjing

网络运行环境（network operation environment） 开放系统应用开发运行环境和网络计算环境两部分组成的集合。前者为各种应用系统的开发和运行提供了统一的支撑平台；后者则在网络范围内将应用开发和运行环境连接在一起。

（1）应用开发运行环境 **应用开发运行环境**实际上是网络中间件（例如，用户接口服务、网络服务、数据管理服务和数据交换服务等）和**操作系统**的组合，目前比较流行的产品有客户信息控制系统（CICS）、通用开放系统环境（COSE）和 Windows 开放服务结构（WOSA）。

目前基于开放源码如 Linux 和 GNU 的应用开发运行环境得到越来越多的认可，成为发展趋势。

（2）网络计算环境 **网络计算环境**主要实现网络范围内的数据管理、通信服务和网络管理。在数据管理方面有用于数据库间通信的远程数据访问（RDA）。通信服务方面有开放系统互连（OSI）、远程过程调用（RPC）以及分布计算环境 DCE、CORBA（参见**分布式计算环境**）等。网络管理方面有分布管理环境（DME）和**简单网络管理协议**（SNMP）、**公共管理信息协议**（CMIP）等。

参考文献

1. Marciniak J J. Encyclopedia of software engineering. 2nd ed. New York：John Wiley & Sons Inc.，2002

2. 胡道元. INTRANET 网络技术及应用. 北京：清华大学出版社，1998 （王晓春 徐明伟）

wangluo zhenting

网络侦听（wiretapping） 第三方隐蔽旁听网络会话的活动，也称网络嗅探（sniffing）、网络监听或网络窃听。早期的电话侦听是通过电话线（wire）的分接（tap）来实现的，而现在的互联网侦听则是通过放置嗅探器（sniffer）来实现的。

网络嗅探器是一种设备或装在主机上的软件，其网卡设置成混杂模式后能捕获所在以太网上出现的所有数据包，并被送交上层协议分析软件进行数据还原，从而解析出明文传输的会话内容，也许包括账号、口令等敏感信息。

存在两种网络嗅探方式。一种是被动嗅探，即直接在共享式（如 Hub 互连）以太网络内部署嗅探器，或利用漏洞先登录到受害主机上，再部署嗅探器。另一种是主动嗅探，一般发生在交换式局域网上，大多需要先采取 ARP 欺骗、虚假 DHCP 服务器、ICMP 重定向以及中间人攻击等技术把数据包导向嗅探器，然后才能实现嗅探。

合法或授权进行网络嗅探，可用于支持网络流量监视、数据包分析、资源利用评估、安全规则执行、故障诊断、数据鉴定或网络取证等网络监管业务。而非法进行网络窃听，会造成用户敏感数据或网络内部状态的泄露，是网络安全的严重威胁之一。

反侦听技术主要包括对侦听的检测与防范。检测技术包括，向可疑点（目的 IP 和 MAC 地址）发送 ICMP Ping 包或 ARP 数据包，根据它是否回音，来判断它是否是嗅探器；也可根据它是否发出反向 DNS 查询来进行侦听检测；或通过蜜罐诱捕而发现某些网络侦听。防范技术包括，采用交换式以太网阻止大部分被动嗅探；通过 IP 碎包或低 TTL 包困阻嗅探；或开辟 VPN、SSH 加密通道或隐通道避免明文传输，主动防范侦听。

参考文献

诸葛建伟. 网络攻防技术与实践. 北京：电子工业出版社，2011 （李芝棠）

wangye

网页（Web page） 用**超文本标记语言** HTML（hypertext markup language）书写的纯文本文件，存放在与互联网相连的计算机上，用统一资源定位符 URL（uniform resource locator）进行标识和存取，通过浏览器进行阅读。

网页中的最基本元素是文字和图片，另外还有动画、声音、视频、程序等。网页通过 HTML 语言规定的格式和标记方法进行描述，确定显示对象的字体、颜色、大小、位置等修饰信息。浏览器对这些标记进行解释并按规定的格式生成大家所看到的网页画面。

网页文件通常是 HTML 格式，扩展名为. html 或. htm，也称为静态网页。随着技术的发展，目前出现了很多不同格式的网页文件，如动态网页，其扩展名为 cgi、asp、jsp、php、shtml 等。

不是每个网页文件都包含 HTML，一个网页可以由 PDF 格式的文档、GIF 格式的图形、JPEG 格式的照片、MP3 格式的歌曲、MPEG 格式的视频或者其他数百种文件类型的任何一种组成。

网页文件中只存放文本、格式描述以及链接信息,网页链接中指向的图片、动画、音频、视频、程序等文件与网页文件是互相独立存放的,甚至可以在不同的计算机上。

每个网页文件都有一个用统一资源定位符URL定义的互联网地址,URL描述了各类文件在互联网中的位置和访问方法。每个网页文件内部也包含多个由URL组成的超链接,指出其他文件的位置以及浏览器如何处理它们。网页中的超链接分为**超文本**(hypertext)链接和超媒体(hypermedia)链接。浏览器一般通过增加下划线、改变颜色等方式突出显示链接,当鼠标移动到这些突出显示的链接上面时,鼠标光标会发生变化。单击这些链接时,就可以根据网页超链接中描述的URL地址跳转到其他文档,实现对其他网页、动画、声音、视频、程序等资源的访问。

由于HTML文件是普通的文本文件,所以只要有一个文本编辑器就可以编写网页文件。但是,用普通的文本编辑器来编写HTML文件并不方便,必须记住所有的HTML标记符号,书写起来非常烦琐,容易出错。现在,有一些工具有助于编写HTML文件,比如Adobe的Dreamweaver、Microsoft的Frontpage以及W3C的Amaya。另外,很多办公软件都具有将其编辑的文件保存为HTML文件的功能。这些工具使用户可以不必了解HTML本身就能够制作网页。网页制作好后,需要发布到Web服务器中,才能被浏览器所访问。

若干内容相关的网页组成的集合称为**网站**(Web site),一个网站的起始网页称为**首页**(home page)。

参考文献

1. 胡道元. 计算机网络. 高级. 北京:清华大学出版社, 1999

2. Tanenbaum A S. 计算机网络. 4版. 北京:清华大学出版社, 2004

(胡道元　张蓓)

wangzhuang shujuku

网状数据库(network database)　采用网状模型的数据库。

网状模型用网状结构表示各类实体及其间的联系。在网状结构中:

(1) 允许一个以上的节点没有父节点;

(2) 一个节点可以有多于一个的父节点。

网状模型中每个节点表示一个记录类型(简称记录型),每个记录类型可包含若干个字段,节点间的连线表示记录型间的联系。记录型描述实体,字段描述实体的属性。网状数据库中记录型之间的联系不唯一。因此,要为每个联系命名,并指出与该联系有关的父记录和子记录。

图1是一个网状模型的例子。

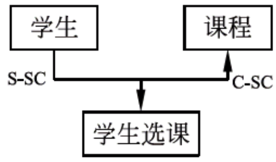


图1　一个简单的网状模型

在此例中,一个学生可以选修若干门课程,某一门课程可以被多个学生选修,学生与课程之间是多对多的联系。引入学生选课的联结记录,从而把学生与课程之间是多对多的联系分解为学生与学生选课,课程与学生选课之间两个一对多的联系。学生与学生选课之间的联系被命名为S-SC,课程与学生选课之间的联系被命名为C-SC,学生和课程为学生选课的父节点。

网状数据库的典型代表是DBTG系统,亦称CODASYL系统,这是20世纪70年代数据系统语言研究会CODASYL下属的数据库任务组DBTG提出的一个系统方案。虽然它仅是一个方案,但它所提出的基本概念、方法和技术具有普遍意义。很多实际运行的网状数据库系统,如IDMS, IDS/2, DMS1100等,都是以DBTG模型为基础,通过对其进行简化和修改而形成的。

DBTG的系统结构由子模式、模式、内模式组成,它是一个典型的三级模式结构。

DBTG是宿主语言系统。用户用宿主语言(如COBOL)编写应用程序,并在需要访问数据库时嵌入数据操纵语句完成对数据库的操作。DBTG网状数据库中各记录型之间的联系用系来描述。数据库中既要储存数据本身,还要反映出数据之间的联系。网状模型中,系通常以链的方式来实现,包括单向链、双向链、环状链、非环状链、向首链等,对系的每一个系值从首记录开始按系序用指针依次连接各属记录。

网状模型是一种比层次模型更具普遍性的结构,它去掉了层次模型的限制,允许多个节点没有父节点,允许节点有多个父节点,此外它还允许两个节点之间有多种联系(称之为复合联系)。因此网状数据模型可以更直接地去描述现实世界。而层次结

构实际上是网状结构的一个特例。

网状数据库虽解决了**层次数据库**存在的一些问题,但是,在网状数据库中,用户对数据库的存取必须沿着存取路径到达目标数据,这就必须随时记录数据库在各个范围中的当前值,加重了用户的负担;对数据的操作是一次一个记录的存取方式,程序和数据虽有较高的物理独立性,但逻辑独立性不高。

网状数据库系统在20世纪70年代与80年代初非常流行,在数据库系统产品中占主导地位,虽然近年来逐渐被**关系数据库**系统取代,但目前在美国、加拿大等国家,由于历史原因,网状数据库的用户数仍然很多。

参考文献

1. 王珊,萨师煊. 数据库系统概论. 4版. 北京:高等教育出版社,2006

2. Date C J. An introduction to database system. 6th ed. Reading: Addison Wesley Publishing Company, 1995

(王珊　周龙骧　王翰虎)

weichengxu kongzhiqi

微程序控制器(micro-programmed control unit, MCU)　通过执行由若干条比机器指令低一层次的微指令所组成的微程序而实现机器指令所必需的各种基本操作的控制器。

微程序设计思想最早是英国的M. V. Wilkes在1951年首先提出的。但是在相当长时间内,由于缺乏快速的微程序存储器而未能推广使用。随着高速半导体**只读存储器**的发展,1964年IBM 360系列计算机成功地采用了微程序控制方案,方便地实现了不同计算机间指令兼容问题,从此微程序控制方案获得广泛应用。

在采用微程序控制器的计算机中,每一条机器指令的执行可以分解为一系列更为基本的操作,称为微操作。控制进行各种微操作的信号称为微命令。根据指令功能的分步要求,将可同时执行的微命令组合在一起,形成微指令。与每一条机器指令相对应的一段微程序是用微指令编写的一段微指令序列。在控制器内部存放微程序的存储器称为**控制存储器**或微存储器,它一般由**只读存储器芯片ROM**或**可编程只读存储器芯片PROM**组成。

微指令分类

微指令可分成水平型微指令和垂直型微指令两大类。

(1) 水平型微指令　由控制字段和下址字段构成如下:

控制字段	下址字段

控制字段中的每一个二进制码位代表一个微命令,用来控制相应的微操作,如某一位指定为"加法微命令",则当该位为1时,控制ALU进行加法运算;为0时,ALU不进行加法运算。下址字段用来决定下一条即将执行的微指令的地址(称为后继微指令地址)。由于复杂指令集计算机的微命令很多,因此水平型微指令的控制字段的长度一般在百余位到几百位之间。这种微指令操作并行度高,编写的微程序短,速度快,使用灵活,适用于大中型高速计算机。缺点是微指令字较长,控存位码利用率不高,形成立即数和转移地址困难。

分段编码法水平型微指令　为了兼顾速度指标与经济性,在满足速度要求的前提下尽量缩短微指令字长,通常采用分段编码法。其原理是将水平型微指令的操作控制字段按照数据通路及微操作的互斥性分成若干小组(称为字段)。每组微命令用若干位二进制编码表示,通过译码器控制相应微操作的执行。各个控制字段译出的微命令可以同时出现,实现并行操作。

(2) 垂直型微指令　其格式与机器指令很相似,通常一条微指令完成一种基本运算或基本操作,如加法微指令、移位微指令、转移微指令等。微指令中设置微操作码,N位微操作码可表示2^N种微指令。微指令中还设置源寄存器地址字段和目的寄存器地址字段,指明操作数的地址。微程序通常是顺序执行的,微程序控制器中还设置微指令计数器。为提高操作并行度,微指令字中还可增加辅助功能字段。基于垂直型微指令的微程序称为垂直型微程序。其特点是:①微指令字较短,控存位码利用率高;②编制微程序容易;③可以采用高级微程序设计语言,容易实现设计自动化。其缺点是操作并行度低,编写的微程序较长,完成一条机器指令时间较长。

毫微程序　通常采用二级控制存储器结构,第一级控制存储器存放垂直型微指令,用以解释指令;第二级控制存储器存放水平型微指令,用以解释垂直型微指令,执行指令过程中由垂直型微指令调用水平型微指令。采用上述结构的微程序称为毫微程序。

微程序的顺序控制

根据现行微指令得出下条微指令在控制存储器中的地址,即后继微指令地址,有以下两种情况:

（1）无分支流程　微指令的后继地址是唯一的。可以设置微程序计数器 μPC，每执行一条微指令，μPC 加 1，形成下一条微指令地址，这是顺序执行的情况。也可以由微指令的下址字段指定下一条微指令地址，实现顺序执行或无条件转移。

（2）分支流程　一条微指令完成后，根据测试条件决定转向哪一条后继微指令。

实现分支流程（条件转移）的方法有多种，如：①用类似机器指令条件转移的方法，产生下一条微指令地址。②在下址字段中，设置若干位较短的修改地址字段，用来指出转移地址的低位部分，形成不同的分支地址。③用测试条件去修改微指令后继微地址字段，形成不同的分支地址。④用可编程逻辑阵列 PLA 给出不同条件下的分支地址。

微程序设计的方法与程序设计类似，可以设计成循环微程序、微子程序与公用微程序等。其中微子程序与公用微程序指可被多个微程序调用的一段微程序。微子程序保存返回微地址，微子程序最后一条微指令应具有返回功能。

微程序控制器的组成

微程序控制器逻辑框图举例如图 1 所示。除了控制器常设的一般部件，包括**程序计数器** PC、**指令寄存器** IR、时序电路和中断逻辑以外，还有保存微程序的控制存储器 CM 以及为执行微程序而设置的各种电路。简述如下：

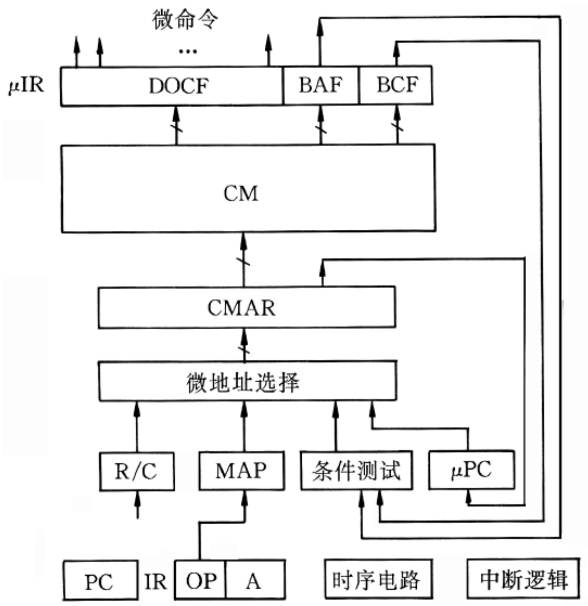


图 1　微程序控制器逻辑框图

控存地址寄存器 CMAR　保存即将执行的微指令在控存中的地址。

微指令寄存器 μIR　包括操作控制字段 DOCF，转移控制字段 BCF 和转移地址字段 BAF。DOCF 产生各种微命令，BCF 给出转移条件，BAF 提供转移地址。通过条件测试电路决定后继微地址，送控存地址寄存器。

机器指令的微程序入口地址由机器指令操作码通过地址映射电路 MAP 得到；当微指令顺序执行时，可通过 μPC 加 1；当执行转移时，可通过条件测试电路决定后继微地址；当执行转微子程序时，返回微地址可保存在 R/C（返回-计数）寄存器中。当执行循环微程序时，在 R/C 中保存循环次数，R/C 有计数功能。

参考文献

1. 金兰，金波. 计算机组织：原理、分析与设计. 北京：清华大学出版社，2006

2. 陈炳从. 电子计算机微程序设计技术. 北京：国防工业出版社，1981

（谢树煜）

weichuliqi

微处理器（microprocessor）　把计算机的**中央处理器**功能实现在单片集成电路或少数几片集成电路中的芯片器件。它是微型计算机的核心部件。

微型计算机的传统结构如图 1 所示，其中虚线框内是微处理器，它包括 3 个基本部件：

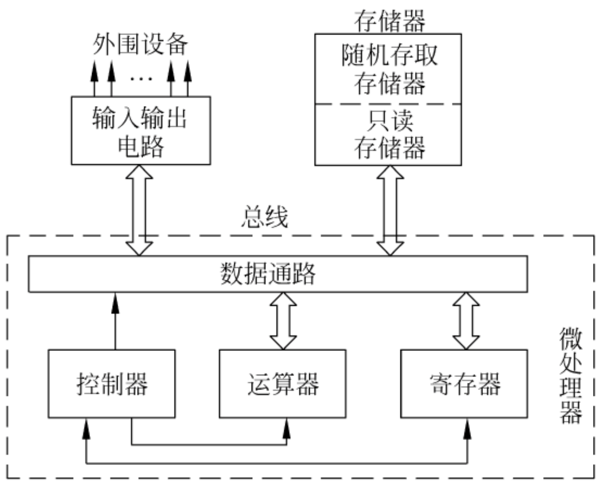


图 1　微处理器与微型计算机的结构框图

（1）**运算器**　它既能执行算术运算，又能执行逻辑操作。

（2）寄存器　每个微处理器中都有多个寄存器，用来存放操作数、中间结果以及标志工作状态的信息等。

(3) **控制器**　它包括控制操作的电路以及用于定时的时钟脉冲发生器等。

自从1971年第一个4位微处理器芯片Intel 4004问世以后,很快就有大量8位微处理器如Intel公司的8080、Motorola公司的6800、Zilog公司的Z80等推向市场。在经历了16位微处理器发展阶段后,32位或64位微处理器迅速推广,并成为目前国际上的主流。Intel公司继1987年推出32位的80386微处理器后,不久又推出80486微处理器,其速度达到54MIPS。1993年,Intel公司推出的Pentium微处理器,采用0.8 μm的BiCMOS工艺,集成了310万只晶体管,在时钟频率为66 MHz时,功耗为13 W。1994年,Intel公司推出时钟频率为100 MHz的Pentium微处理器。1995年又推出时钟频率为150 MHz和166 MHz的Pentium Pro微处理器。以后,Intel把Pentium Pro改称为Pentium Ⅱ,并推出Pentium Ⅲ和Pentium 4。Intel把这一系列32位字长的微处理器芯片称为IA32体系结构。2000年,Intel与HP合作,又联合开发研制出64位的微处理器芯片Itanium,称为IA-64体系结构系列,主要用于服务器产品市场。另一方面,除了用于台式计算机和笔记本计算机的微处理器以外,近年来,用于移动式计算机(包括平板计算机)的移动微处理器发展也很快,这种用途的微处理器要求工作电压低、省电,计算能力要求不太高,而数据处理和图形处理能力要求较强。

后来,微处理器芯片性能的发展,受到了片上功耗的限制,即其芯片性能的提高,不能像过去依靠不断提高芯片工作频率来实现,频率的提高会使芯片功耗高而发热到不可容许。然而,微电子工艺仍然继续发展,芯片上可容纳的晶体管数仍在不断增加。为了解决芯片的性能和功耗的矛盾,一方面采用较低的工作电压,以减低功耗;另一方面在压低工作频率的同时,发展了片上多核(multi-core)和众核(many core)的技术来提高性能,*多核*一般是从二核到八核,*众核*则从几十核到几百核,芯片上的多个同构或异构的核之间则用片上内部的互连网络来连接。这实际上已经是在芯片上实现多机系统,称为片上系统SOC(system on chip);而片上内部互连网络则称为片上网络NOC(network on chip)。这种芯片,已经不能再用微处理器来称呼,因为它还包含了多级片上存储和各种总线,所以称为微处理机,有时就直接称为中央处理机芯片(CPU chip)。AMD公司的产品与Intel产品相兼容,并且与Intel公司进行了竞争,研究和发展Opteron和Athlon等多核系列芯片。目前国际上工业大规模生产的微电子工艺已做到32 nm,主频可达3 GHz左右,片上晶体管数达几十亿个。

现代微处理机与80年代的80x86微处理器有很大的不同。(参见"**处理机体系结构**")主要方面有:①从每个周期发射一条指令,发展到发射多条指令;使多条指令尽可能地并发执行,成为微处理机设计考虑的重要问题。②指令的执行多采用流水线操作。因此,现代微处理机是超标量和超流水线体系结构的,它采取了多种措施提高指令执行的并行性。③现代微处理机一般是多核的,并且都带有片上的高速缓存,包括指令高速缓存和数据高速缓存以及包括第二级高级缓存。这些高速缓存在片上的晶体管数,往往占片上晶体管总数的50%~60%。

参考文献

1. 李三立. 微处理器与微型计算机. 北京:国防工业出版社,1981

2. 李三立. RISC——单发射与多发射体系结构. 北京:清华大学出版社,1994

3. Gilmore C M. Microprocessors: principles and applications. McGraw-Hill Publication,1995

(李三立)

weikongzhiqi

微控制器(microcontroller)　由**中央处理器**、存储器、输入输出端口(包括并行I/O、串行I/O、模数转换器)、计时器和计数器等组成,具有完整数字处理功能的大规模集成电路。微控制器是一种面向控制领域嵌入式应用的集成化计算机芯片,主要用于工业控制、数据处理、信号处理、智能仪器、通信产品及民用消费产品等自动控制产品与器件中。通常也把它简称为MCU或μC。MCU配以适当的外围设备和软件就可构成一个计算机应用系统,所以也称之为*单片微型计算机*,简称为*单片机*。

MCU的发展始于20世纪70年代中期,当时主要称为单片机。由于工艺和集成度的限制,一个完整功能的MCU由两块集成电路组成,如Fairchild公司的单片机F8必须外接一块专为F8设计的程序存储单元电路3851。第二阶段为低性能MCU阶段,虽已只用一块芯片构成,但性能低,品种少。如Intel的MCS-48系列,芯片内含有中央处理器(CPU)、并行I/O口、计时器、随机存取存储器(RAM)和只读存储器(ROM)等,但其CPU功能不强,I/O的种类

和数量少,存储容量小,只能应用于要求比较简单的场合。第三阶段是高性能微型计算机系统(MCS)发展阶段。此时的MCU内部具有功能很强的CPU、比较多的输入输出电路和大容量的数据存储器、程序存储器。MCU产品型号、规格多,各具特色,能满足不同领域应用需求。

MCS分为通用型和专用型两大类。通用型MCS是把可开发资源(如ROM,I/O口等)全部提供给用户,以下所讨论的MCS单指通用型MCS。专用型MCS实际上是一种微控制系统集成化的产品,如频率合成调谐器、打印机控制器等。按MCS基本操作处理的数据位数区分,可有1位MCS、4位MCS、8位MCS、16位MCS和32位MCS等。

随着大规模及超大规模集成电路技术的发展,微控制器的集成度也在不断提高。作为一个自成体系的单片微型计算机,除了CPU之外,微控制器还包括超高速缓冲存储器、主存储器、输入输出接口、直接存储器存取(DMA)控制器、中断处理器、计时器以及其他高性能微控制器所必需的子系统。同时,微控制器具有丰富的输入输出设备,例如时钟发生器,输入输出串行端口以及其他串行通信接口,比如串行外设接口(SPI),控制器区域网(CAN)等,许多微控制器具有模数转换器(ADC),有的还带有数模转换器(DAC),等等。通常,这些集成在内部的设备可以通过特殊的指令来操作。

微控制器正朝着高性能和多品种的方向发展。MCU性能的提高主要体现在下述4个方面:①增强MCU内部CPU功能,即提高其数据处理速度和精度。包括扩大字长,增加乘法、除法部件,采用流水线结构等。有的MCU则将高性能16位或32位微处理器原封不动地移来,作为MCU的CPU单元,这样MCU便与相应的通用计算机系统的软件兼容,具有相同的数据处理能力,便于开发和应用。②增加MCU内部资源的集成,尽量减少外接电路,使MCU本身即是一个完整的应用系统。例如增加内部存储器种类及容量,扩大I/O端口品种和数量。I/O端口可以有串行口、并行口、多路模数转换器、计时器、定时输出和捕获输入、系统故障监视器、DMA通道及数模输出电路等。③设计和使用多功能复用引脚,使MCU内部资源的增加不会导致MCU外引脚的过多增加,从而提高应用的灵活性。④扩大外寻址范围以提高系统的扩展功能。目前外寻址空间已从典型的64 k字节扩大到几兆字节。MCU品种的发展趋势是多品种和多层次,主要体现为:①发展低电压和低功耗品种,如电池供电系统和野外作业系统。②发展微型化MCU,即利用MCU设计模块化的结构特点,在内核CPU不变的情况下根据应用目标调整内部各功能模块的规格和外引脚数,以发展新品种。

目前,一些微控制器的性能已可与主流微处理器相媲美,其应用已经渗透到工业和管理的各个方面。在机器人、机械工具、汽车、飞机、宇宙飞船、医疗电子设备等许多仪器和装置上,微控制器都起着极其重要的作用。

参考文献

1. Hintz K J, Tabak D. Microcontrollers: Architecture, implementation, and programming. New York: McGraw-Hill Inc., 1992

2. Peatman J B. Design with microcontrollers. New York: McGraw-Hill Inc., 1988

3. Wilmshurst T. Designing embedded systems with PIC® microcontrollers: Principles and applications. 2nd ed. Oxford: Newnes/Elsevier Ltd., 2009

(马玉海)

weineihe

微内核(microkernel)　**操作系统**中仅包含为所有应用所必需的资源控制与通信功能的内核。内核是操作系统中最靠近硬件且享有最高特权的一层。微内核功能以外的功能作为服务程序在内核外实现,供使用时调用,这样,可使操作系统之内核为最小。

20世纪70年代,随着操作系统的功能日益增强,系统日益复杂庞大,其内核亦随之增大,从而出现了微内核概念。其优点是使操作系统易于理解、实现、维护和移植,系统服务剪裁与配置较为灵活,利于适应不同应用的要求,但是,置于内核外的系统服务效率将会下降,含于内核内的功能的灵活性受到限制。因此,在设计具微内核结构的操作系统时重要的考虑因素是性能与灵活性之间的权衡以及易维护性与系统开销之间的权衡。

为使在不具微内核结构的操作系统上运行的应用程序也能在具微内核结构的操作系统上继续使用,通常在具微内核的操作系统与应用程序之间提供一个针对不具微内核结构的操作系统(如UNIX)的仿真接口,将应用程序的系统调用转换成对具微内核结构的操作系统的调用。

美国卡内基梅隆大学开发的Mach是一个著名

的具微内核结构的操作系统。Mach 与 UNIX BSD 完全兼容。但 Mach 3.0 采用一个很小的内核,将 **UNIX 操作系统**功能都移到内核以外,并实现了多种仿真接口。例如,Apple MacOS X 服务器操作系统也采用了 Mac 内核。

Microsoft Windows NT(参见 **Windows 操作系统**)可以看作一个混用层次结构与微内核结构的例子。Windows NT 可以运行 Win32、OS/2、POSIX 等应用,其内核支持客户应用与应用服务器之间的消息传递,而在内核以外运行针对各种应用类型的服务器。

参考文献

1. Bacon J. Concurrent systems—operating systems, database and distributed systems: an integrated approach. 2nd ed. Addison-Weisley, 1998

2. Silberachatz A, Galvin P B, Gagne G. Operating system concepts. 6th ed. John Wiley & Sons, Inc., 2002

(陈道蓄)

weixing jisuanji

微型计算机(microcomputer) 以**微处理器**为**中央处理器**而组成的**计算机**系统。又称微型机,或简称微机。传统的微型计算机由微处理器芯片、**半导体存储器**及其他辅助电路安装在印刷电路板上,再配置必要的外围设备组成。有的微型计算机只有一块电路板,称为单板微型计算机,或简称单板机。随着微电子技术的发展,根据应用需要微型计算机电路可以做在一个芯片上,称为**单片计算机**,或简称单片机。

发展简史

微型计算机的历史是从 1971 年美国 Intel 公司推出 4004 微处理器开始的。按处理数据的通路宽度和功能,微型计算机的发展大致可以分为 5 个阶段。

第一阶段(1971—1973) 典型的微型计算机以 Intel 公司的 4 位微处理器 4004 和 4040 为基础。微处理器和存储器采用 P 沟道金属氧化物半导体(PMOS)工艺,工作速度很慢。微处理器的指令系统不完整;存储器的容量很小,只有几百字节。微型计算机中没有操作系统,只有汇编语言。这种微型计算机主要用于工业仪表、过程控制或计算器中。

第二阶段(1974—1977) 代表性的微型计算机以 8 位微处理器为基础,典型的微处理器为 Intel 公司的 8080 和 8085、Zilog 公司的 Z80 及 Motorola 公司的 6800。微处理器采用 N 沟道金属氧化物半导体(NMOS)工艺,具有较完整的指令系统和较强的功能;存储器容量达 64KB,配有荧光屏显示器、键盘和软磁盘等输入输出设备,构成了独立的**台式计算机**。在微型计算机中配备有简单的磁盘操作系统(如 CP/M)和高级语言,这种台式微型计算机又称为**个人计算机**,就是迄今仍广泛使用的名词 PC。实际上,凡是最初设计来为用户独自使用其计算能力和各种媒体处理能力的微型计算机,都称为个人计算机。

第三阶段(1978—1981 年) 代表性的微型计算机以 16 位和准 32 位微处理器为基础构成,其典型的微处理器有 Intel 公司的 8086、Motorola 公司的 68000 和 Zilog 的 Z8000。微处理器采用短沟道高性能 NMOS 工艺;在体系结构方面,吸收了传统的小型计算机甚至大型计算机的设计思想,如虚拟存储和存储保护等。这时的微型计算机已有相当强的功能,存储容量可达 1MB,还可配备较大容量的软磁盘和硬磁盘。在这一阶段,操作系统、高级语言、工具软件和应用软件也日益成熟、丰富。在此期间,多用户微型计算机系统、多处理机微型计算机系统已开始出现。工业控制微型计算机等也得到了发展。

第四阶段(20 世纪 80 年代初期至中期) 80 年代初,IBM 公司推出开放式的 IBM PC 是微型计算机发展的一个里程碑。IBM PC 采用了 Intel 80x86(当时为 8086,80286 和 80386)微处理器和 Microsoft 公司的 MS-DOS 操作系统,IBM 公司还公布了 IBM PC 的总线设计。IBM 在硬件、操作系统和总线接口这三个方面的开放,为微型计算机的大规模生产打下了基础。各国很多公司纷纷研制与 IBM PC 兼容的微型计算机及其配套的板级产品和外围设备,很多软件公司分别研制和开发在 MS-DOS 基础上的软件。当时,IBM PC 所用的芯片、操作系统和总线实际上形成了国际性的工业生产的主要标准,从而使微型计算机的生产发展成为大批量规模性经济的产业,推动了微型计算机应用的飞速发展。与此同时,美国 Apple 公司推出的微型计算机具有菜单式的选择功能和窗口图形用户界面,使微型计算机的人机交互使用更加方便。

第五阶段(从 20 世纪 80 年代中后期开始) RISC(参见**精简指令集计算机**)技术的问世使微处理器的体系结构发生了重大的变革。RISC 微处理器的设计周期短,工作速度快。在 1987 年 RISC 微型计算机进入批量生产时,其运算速度已达每秒几

千万次,后来的 RISC 微型计算机可达每秒几亿次,并且可以采用 UNIX 操作系统。Intel 由于已经有"80x86"微处理器发展的基础,仍然坚持发展 x86 为二进制兼容核心的 CISC(参见**复杂指令集计算机**)微处理器,并大量采用了 RISC 的技术优点,使微处理器芯片的功能飞跃发展,尤其是多核(一般 2 核到 8 核)和众核(几十到几百核)中央处理器芯片技术的实现,使这种芯片已经实际上实现了很强功能的多机系统,其功能已经超过过去小型计算机、甚至大型计算机的中央处理机,而价格却由于芯片大规模生产而更加便宜。这种变革使微型计算机、小型计算机和大型计算机的界限越来越模糊,从而使传统上的小型计算机和大型计算机的组成技术也发生巨大的变化。同时,这种变革也促使计算机工业发生深刻变化,很多有名的生产小型计算机和工作站的公司倒闭。以微型计算机为节点的多机系统迅速发展,已成为较强功能的服务器、功能强大的主机和超级计算机系统的基础。

分　类

微型计算机分类的方法很多。从处理数据宽度来分类,可分为 8 位微型计算机、16 位微型计算机、32 位微型计算机和 64 位微型计算机。从组装形式来分类,微型计算机可以分为便携式微型计算机和非便携式微型计算机。非便携式微型计算机又可分为台式计算机和塔式计算机;便携式微型计算机是一种可移动的微型计算机(参见**移动式计算机**)。最初的便携式微型计算机具有较大的显示屏,连同磁盘和键盘可全部装入一个手提箱中,称为手提式计算机。早期的便携式微型计算机分为膝上计算机、笔记本计算机和掌上计算机。后来膝上计算机趋于淘汰,掌上计算机现在发展成为广泛采用的"**移动终端**"。便携式微型计算机的体积向更小的方向发展,而且人机交互的输入和输出方式更加多样化,如触摸屏输入、语音输入和输出、笔输入等等。便携式计算机又分出平板式计算机,它是把平板液晶显示器和 CPU、存储器和其他部件都装配在一个平板中,携带和使用更方便。平板计算机生产的数量日愈庞大。

根据微型计算机是否由最终用户使用,微型计算机可分为独立式微型计算机和嵌入式微型计算机(参见**嵌入式计算机**)。独立式微型计算机由最终用户直接使用,最常见的是个人计算机。嵌入式微型计算机作为一个信息处理部件装入一个应用设备中,最终用户不直接使用计算机,使用的是该应用设备,例如包含有微型计算机的医疗设备、仪器设备、摄像机等。嵌入式微型计算机一般是单片机或单板机。根据用途来分类,微型计算机可分为通用微型计算机和专用微型计算机;也可分为民用微型计算机、工业用微型计算机和军用微型计算机。工业用微型计算机和军用微型计算机对于环境适应能力、抗干扰能力等的要求比民用微型计算机的高,因而采用加固的组装结构,而且往往要求有中断响应快的实时操作系统。

微型计算机的内部结构

微处理器是微型计算机的核心部件,它的基本组成部分包括:运算部件、寄存器组、控制部件以及由数据线、地址线和控制信号线组成的内部总线。微处理器能够完成取指令、指令译码、取操作数、执行指令、送结果等操作以及按状态字执行特定操作,例如处理异常事件、执行与外围设备交换信息的操作等。微型计算机的存储部件包括随机存取存储器、只读存储器以及相应的读写电路和控制电路。现在一般微型计算机包含高速缓冲存储器;还包含存储管理部件 MMU,它主要用于分配存储空间,并保护某些特定的存储空间,防止不合法的访问。存储管理部件可以与微处理器集成在同一芯片上,也可以单独做成一个芯片。一般的微型计算机包括输入输出接口和常规的外围设备,即键盘、鼠标器、荧光屏显示器、打印机、软磁盘和硬磁盘。单板机的输入设备可以是小键盘,输出设备可以是发光管或液晶显示屏。仪器或工业控制用的微型计算机输入和输出一般需要模数和数模转换设备。新型的微型计算机还可配备多种通信设备和语音、图像或手写体文字的输入输出设备等。

发展趋势

应用最广、产量最高的两种微型计算机是个人计算机和单片机。单片机已经广泛用于家电、生活用具和仪器仪表,正在向智能化发展。个人计算机除了处理数据、文字外,还可成为处理图形、图像、语音和声音的多媒体计算机;在 3G/4G 通信时代,还要处理流媒体数据。个人计算机和通信更紧密地结合并作为网络的终端机(包括有线网络和无线网络终端)是微型计算机的一个重要发展趋势。从结构来说,平板计算机的生产数量可能超过非平板计算机。多台微型计算机并行工作,则可以实现性能价格比高的高性能计算机系统。

参考文献

1. 李三立. 微处理器与微型计算机. 北京:国

防工业出版社,1981

2. 张福炎,等. 微型计算机 IBM PC 的原理与应用. 南京:南京大学出版社,1984　　　　（李三立）

weizuzhuang jishu

微组装技术(micro packaging technology)
以微电子、高密度组装和微焊接等技术为基础,在多层布线基板上,将微电子器件及微型元件组装成电子硬件的一种工艺技术。它涉及固态技术、薄膜技术、厚膜技术、微电路技术、互连与微焊接技术、热控制技术、高密度组装技术、测试技术、可靠性技术和计算机辅助工程等领域,是一门电路、结构、工艺、材料、元器件等紧密结合的综合性技术。

电子组装技术经历了下列几个阶段:①20 世纪 40 年代是以电子管为有源器件的手工焊接阶段。②40 年代晶体管和印制电路相继问世,并在 50 年代至 60 年代得到广泛应用后,形成了以晶体管和印制电路板为主的手工焊接阶段。这阶段电子设备的组装和结构产生了很大的变化,提高了组装密度,缩小了设备的体积。③60 年代,在集成电路技术与多层印制板发展的基础上,形成了以集成电路、自动插装和**波峰焊**为主的组装阶段。④70 年代末,由于超大规模集成电路和无引线或短引线片状电子元器件的发展,电子组装进入了表面安装阶段。⑤80 年代中期,在发展**表面安装技术**的同时,微焊接技术、高密度多层基板技术的发展,形成了以**多芯片模块**(MCM)为特征的第五代微电子组装阶段。其特点是组装密度更高,互连线更短,因此,信号延迟时间短,信息传输速度快。随着组装工艺及材料科学的不断发展,微组装又向三维立体组装发展。目前三维组装主要有两种形式:一种是在氧化铝基板上先形成薄膜电路,然后敷上一层聚酰亚胺薄膜,再在此薄膜上镀镍或铜,以形成带状薄膜电缆所需的布线图形,而后用再流焊将倒装片集成电路、片式电容器和片式电阻器等焊在此薄膜电缆上。另一种形式是将电容和电阻等埋在多层陶瓷基板的内部,利用同时烧成技术,使薄膜电缆、电子元件埋藏在基板的层间。三维组装芯片间的互连线更短,因此,减小了芯片间的阻容负载和信号延迟时间,也减小了寄生效应。

微组装与常规电子组装的主要区别在于所用的组装结构和互连技术不同。常规的电子组装是以一般电子元器件及普通印制电路板为基础的组装技术,微组装则是以集成电路和高密度多层基板以及微焊接为基础的综合性组装技术。微组装能减小电子元件和芯片的安装面积及互连线的长度,并能扩大基板尺寸和布线层数,以容纳更多的集成电路芯片和元器件,从而提高其组装密度,完成更多更复杂的电路功能。

高密度多层布线技术是微组装中缩小组装器件(或设备)的体积,减小布线总长度,缩短信号延迟时间,提高信息传输速度的一项关键技术。一般采用薄膜和厚膜技术布线,要求布线结构微细,布线长度尽可能地短,还要求线间的介电常数小。通常有下列 4 种布线技术。

(1) 多层陶瓷布线　在低温烧结玻璃陶瓷多层布线基板上,用光刻技术形成金或银钯薄膜布线。在底层基板上,用生带技术形成电源层和接地层布线,而上层基板上则是信号层布线,在各层陶瓷表面上溅射钛、钯两层膜,经光刻、显影形成布线图形。

(2) 多层薄膜聚酰亚胺布线　在陶瓷或硅基板上电镀或溅射 Cr/Cu/Cr 电源层,敷上聚酰亚胺薄膜,在其上再涂光致抗蚀剂,并制作通孔图形,以抗蚀剂为保护膜,腐蚀聚酰亚胺,形成互连通孔。除去抗蚀剂后,在表面溅射几微米厚的 Cr/Cu/Cr 层,用光刻法形成第一层信号布线层,再涂敷介质聚酰亚胺层,形成互连通孔,然后按上述方法形成第 2 信号布线层和接层等。这种布线技术已达 63 层。

(3) 厚膜布线　用微型笔或浆料喷射法,直接在厚膜导体层上按设计要求形成微细布线图形,布线材料一般用铜。

(4) 混合布线　一种厚、薄膜结合的布线技术,用标准厚膜工艺形成电源层和接地层布线,用光刻光形成互连通孔,再在上面用薄膜工艺形成信号层布线。这种布线技术的组装密度高于上述几种布线技术,陶瓷基板尺寸可减小几倍,适用于高频多芯片模块的组装,能降低信号衰减和串扰,成本也比较低。目前国际上正在研究利用超导技术进行布线,超导布线可将 40～60 层布线减小到只有 4 层布线(电源层、接地层、x 层和 y 层)结构。

改进芯片及元器件的互连和安装方法是缩小体积、提高组装密度和可靠性的一项重要途径。目前在微组装中常用的微焊接方法有丝焊、激光微焊、芯片基板焊、倒装焊(即**倒装芯片焊接**)和载带自动焊等。丝焊和倒装焊属直接安装法,安装面积小,但芯片不能老化筛选和预测,影响混合电路或微电子组装组件的合格率和可靠性。芯片基板焊是一种将集成电路或芯片直接安装与互连到印制电路板上的焊

接技术。载带自动焊是焊接集成电路内、外引线的一种自动群焊法,它先把载带(镀锡或镀金的铜箔、黏接剂塑料膜制成的具有引线框的柔性印制电路板)用热脉冲焊或超声热压焊焊到芯片焊区的镀金凸台上(称内引线焊接),冲剪下此载带,并迅速将其焊到基板的焊盘上(称外引线焊接),整个焊接过程自动完成。

由于微组装的组装密度很高,其体积功率密度(或面积功率密度)也很高,采取适当的热控制技术亦是微组装的一项重要内容。热控制的目的是为了尽量减小电子组件或设备的内部热阻和外部热阻,利用热传导、对流换热和辐射换热把热源的热量迅速散发至周围环境。常用的冷却方法有:自然冷却(包括传导冷却、自然对流冷却和辐射换热)、强迫空气冷却、液体冷却、蒸发冷却、热电致冷(半导体致冷)、热管传热等。冷却方法的选择主要取决于元器件或设备的发热功率密度及其允许的温升。采用散热性能好的基板材料和介质材料,改进元器件内部各电极的焊接质量,均可减小其内部热阻。

参考文献

Sinnadurai F N. Handbook of microelectronic packaging and interconnection technologies. Scotland: Electrochemical Publications, 1985 (赵惇殳)

weihu guocheng

维护过程(maintenance process) 软件维护人员所负责的一系列活动,其目的是在保持软件整体性能的同时对它进行修改,使它达到某一需求,直到其退役才告终止。从维护方式上讲有三种维护:改正性维护、适应性维护以及改善性维护。当软件由于错误、缺陷、问题需要改进和修改,以及对相应文档进行修改时,都要涉及维护过程。

维护过程包含的活动有:问题分析和修改分析、修改和实施、对维护的评审和验收、移植、软件退役等。维护过程同时还贯穿软件过程中其他过程的实施:维护人员通过**管理过程**管理维护过程;通过**剪裁过程**剪裁维护过程中的活动;通过改进过程参与维护过程的管理(参见**软件过程**);通过**支持过程**实施维护过程中的文档编制、评审、质量保证等。当进行某个活动时,维护过程可能需进入开发过程(如在实施软件的修改时),这时维护人员即是那里的开发人员(参见**开发过程**)。

以下是维护过程所涉及的活动,其中活动(1)是必需的,且是应当首先完成的。活动(2)至活动(6)为可选的,开发人员应当首先完成活动(1)的任务,然后根据活动(1)所产生的结果选择活动(2)至(6),完成维护过程。

(1) 本过程的实施准备 目的是为维护过程准备最基本的约定。其主要任务有:①制定活动(2)至活动(6)的实施计划和步骤;②确定对用户的问题报告和修改请求进行接收、跟踪的步骤以及向用户反馈信息的方式和步骤;③指定文档编制方式、配置管理方式以及各支持过程的实施方法。

(2) 问题分析和修改分析 目的是分析软件修改将对系统、接口带来的影响并确立修改方案。主要任务有:①分析维护类型,是改善型、改正型,还是适应新环境型的维护;分析维护的范围,包括修改规模、成本、时间;分析维护对关键问题(如性能、保密性)的影响;②在分析的基础上,选择实施修改的方案。

(3) 修改的实施 目的是对问题及修改进行更为详细的分析,并实施修改。主要任务有:①详细分析问题报告和修改请求,决定哪些文档、软件单元和版本需要修改;②实施修改。此时维护人员需进入**开发过程**完成修改。开发过程中的需求应作出相应的修改,最低要求是保证未经修改的需求不受影响,而新的修改过的需求得到完全、正确的实现。

(4) 对维护进行评审和验收 此活动任务是:维护人员同授权修改的机构一起进行评审,评定经过修改之后的系统的整体性能。

(5) 移植 目的是将一个系统或软件从一个旧的运作环境移植到一个新的运作环境中,并保证该移植活动的正确性。由于涉及软件的修改,因而也是维护活动。其主要任务有:①制定移植计划并执行该计划。计划中至少包括:对需求分析和移植的定义;移植工具的开发问题;软件和数据的转换问题;移植计划的执行问题;移植的验证问题;以后对旧环境的支持问题等;②向用户通告移植计划和移植执行情况;③旧环境和新环境最好并行运行,并向用户提供必要的培训;④对移植活动及移植后的结果进行评审。

(6) 软件退役 软件将根据所有者的要求退役。此时,维护人员应当:①制定软件支持的撤销计划,并执行该计划。计划中至少包括:在多长时间后全部或部分地停止软件支持;系统及有关文档如何存档;若以后仍需要支持时的责任问题;若需要,转移到新的软件上的问题。②向用户通告退役计划及执行情况。③将退役软件的所有文档、记录、数据

归档。

参考文献

1. IEEE Standard for Developing Software Life Cycle Processes — IEEE Std. 1074—1991. 1991

2. ISO /IEC 12207：1995 Information Technology— Software — Part 1： Software Life -Cycle Process. 1995 （宿为民）

weiyena kaifa fangfa

维也纳开发方法（Vienna development method，VDM） 20 世纪 70 年代由 IBM 维也纳实验室的 C. Jones 和 D. Bjorner 提出的，最初是作为描述程序设计语言**指称语义**的元语言，后来发展成支持程序开发的形式化方法。

VDM 基于集合论，其出发点是用一阶谓词（断言）来描述程序的状态空间（程序状态即程序中所使用的全体非局部变量的取值）。VDM 提供了四个基本类型：集合、复合、映射和序列，作为书写规约的基础。每个基本类型都具有相应的运算和关系，可以在断言中使用。程序的功能由刻画其初始状态的前断言及刻画其终止状态的后断言规定。由于程序的功能描述往往涉及某些变量在其执行后与执行前的取值之间的关系，在后断言中可以用特定的符号来引用变量在程序执行前的初值。VDM 采用“面向模型”的方法来描述数据类型（参见**形式规约**）。一数据类型的规约由三要素组成：①状态集；②初始值；③各运算的描述。每个运算均用前、后断言描述。

例 先进先出队列的 VDM 规约

```
Queue = seq of Qel
q0 = [ ]

ENQUEUE(e:Qel)
ext wr q: Queue
post q = ⃖q ⌢ [e]

DEQUEUE ( ) e: Qel
ext wr q: Queue
pre q ≠ [ ]
post ⃖q = [e] ⌢ q
```

这个例子利用序列来给出队列的规约。其中[ ]表示空序列。[e]是仅含 e 的单元素序列，⌢连接两个序列。保留字 pre，post 分别引出前、后断言。出现在后断言中的 ⃖q 表示变量 q 在程序开始执行时的值（即前断言中的值）。保留字 ext wr 表示随后的变量是外部的，其值在本程序中被引用（r）且被修改（w）。施加于 Queue 上的运算有两个：ENQUEUE 和 DEQUEUE。ENQUEUE 不带前断言，表示其前断言为真。

VDM 提供了一套变换规则。按这些规则可以将规约精化，逐步转换为可执行程序，并保证最终得到的程序满足起初的规约。

VDM 的提倡者十分重视将其应用于工业规模的软件开发实践。他们与工业界联合举办 VDM 研讨班，训练系统设计员和程序员掌握、使用这种方法。设计国际标准化组织（ISO）规约语言 VDM-SL 的工作目前正在进行之中。

参考文献

Jones C B. Systematic software development using VDM. New York：Prentice - Hall，1986 （林惠民）

wei suijishu

伪随机数（pseudo-random numbers） 在数字计算机上用数学方法产生的、具有[0，1]区间上均匀分布母体性质的数值序列 $\{r_n : n = 1,2,\cdots,0 \leqslant r_n \leqslant 1\}$。在一台 b 进制、尾数字长为 k 位的计算机上，任取 m 个整数，$x_1,\cdots,x_m$ 作为初值，按某种递推公式 $X_{n+m} = G(x_n,\cdots,x_{n+m-1})$，便可产生一个数列，如果经过统计假设检验，表明这个数列具有[0，1]均匀分布母体的样本性质，就称其为[0，1]均匀分布的伪随机数列。这里“伪”的意义是，它本身并不是真正的随机数，而且总有周期性。我们希望在满足统计性能的前提下，周期尽可能地长。平方取中法是产生伪随机数列最早使用的方法，它将一个 $2s$ 位十进制随机数，平方之后截取中间的 $2s$ 位作为新的随机数，重复上述过程便得到一个伪随机数列。但按此法所产生的伪随机数列有退化的危险，即出现的数字都变为 0 或者形成重复周期变化的序列，且周期难以确定。此外有取中法和移位法，但结果不理想。目前产生伪随机数列比较好的方法是同余法，包括加同余法、乘同余法和混合同余法。

加同余法的公式为：$x_{n+2} = x_n + x_{n+1} \pmod M$，$r_{n+2} = x_{n+2} / M$。但该数列的相关系数太大（约为 -0.3），不宜直接作为伪随机数列。

混合同余法的公式为

$$x_{n+1} = \lambda x_n + c \pmod M,$$

$$r_{n+1} = x_{n+1} / M, \quad n = 0,1,2,\cdots$$

其中 x_0 为初值, λ 为乘子, c 为增量, M 为模。$c=0$, 即为乘同余法。一般取 $M=b^k$, x_0 与 c 的选取对数列的性质有很大的影响。用数论的方法可证明:混合同余数列达到周期 M 的充要条件是: ① c 与 M 互素; ②对每一个 M 的素因子 p, $\lambda-1$ 为 p 的倍数; ③若 M 是 4 的倍数,则 $\lambda-1$ 是 4 的倍数。若 $b=2$, $M=2^k$, 应取 $\lambda=4q_1+1$, $c=2a_1+1$, x_0 为非负整数,其中 q_1, a_1 为正整数。由上述可见,乘同余法不能达到最大周期 M, 但可以证明乘同余法最大可能周期为 $2^{k-2}(k>2)$。如取 $\lambda=8a+3$, $x_0=2b+1$(a, b 为任意正整数),且 λ 的二进制表示中 0,1 的出现不规则时,便可得到周期为 2^{k-2} 且统计性质较好的伪随机数 $\{r_n=x_n/2^k, n=1,2,\cdots\}$。

通过[0,1]均匀分布的伪随机数 $\{r_n\}$, 便可得到各种不同分布的伪随机数。事实上有定理:设随机变量 η 的分布函数 $F(x)$ 连续,且反函数 $F^{-1}(x)$ 存在,则 $R=F(\eta)$ 便是[0,1]上均匀分布的随机变量。因此, $\{\eta_n=F^{-1}(r_n): n=1,2,\cdots\}$ 为 η 的抽样序列。例如 η 为指数分布, $F(x)=1-e^{-\lambda x}$ $(x>0)$, 则 $\{\eta_n=-\frac{1}{\lambda}\ln(1-r_n)\}$ 便是 η 的抽样序列。

如果 η 为 $N(0,1)$ 分布,由于 $F(x)=\int_{-\infty}^{x}\frac{1}{\sqrt{2\pi}}\times e^{-(x^2/2)}dx$ 的反函数没有显式表示,所以不能用上述方法求 η 的抽样序列。通常采用两种方法: ①由中心极限定理, $\eta_n=\left(\sum_{i=1}^{n}r_i-\frac{n}{2}\right)\Big/\sqrt{\frac{n}{12}}$ 有渐近正态 $N(0,1)$ 分布。一般取 $n=12$, 则可用 $\{\eta_n: n=1,2,\cdots\}$ 作为 $N(0,1)$ 的抽样序列; ②坐标变换法:设 r_1, r_2 为两个相互独立的[0,1]均匀分布的随机变数,作变换

$$\eta_1=(-2\ln r_1)^{1/2}\cos(2\pi r_2)$$

$$\eta_2=(-2\ln r_1)^{1/2}\sin(2\pi r_2)$$

则可证明 η_1, η_2 是两个独立的 $N(0,1)$ 分布的随机变数。

产生其他分布随机序列的方法还有舍选法、离散近似法、复合抽样等。但不管用什么方法产生的伪随机数列,都必须进行统计检验以确认它们有良好的统计性质。这些检验有参数检验、分布均匀性检验、独立性检验等。

参考文献

1. 中山大学数力系. 概率论与数理统计(下册). 北京: 人民教育出版社,1980

2. 徐钟济. 蒙特卡罗方法. 上海: 上海科学技术出版社,1985

(金治明)

weizhi genzongqi

位置跟踪器(location tracking devices) 用于动态、实时提供目标物空间位置(X, Y 和 Z 笛卡儿坐标)和方位(俯仰角、偏航角、滚动角)6 自由度信息的设备。该设备可广泛应用在**虚拟现实**、增强现实以及生物医学等领域的研究中,是测量运动范围和肢体旋转的理想选择。

根据测量原理的不同,位置跟踪器可分为电磁式、声学式、光学式、机械式等,不同原理的设备各有其优缺点,主要体现在定位精度、实时性、使用方便程度、可捕捉运动范围大小、成本、干扰性、多目标捕捉能力等方面。

1. 电磁式

电磁式跟踪器是目前应用十分广泛的一类方位跟踪器。电磁式跟踪器利用三轴线圈发射低频电磁



图 1　电磁式位置跟踪器

场,用固定在被测对象上的三轴磁探测器探测磁场的变化信息,利用电磁发射信号和感应信号之间的耦合关系确定被测对象的方位。一般由发射源、接收传感器和数据处理单元组成。

电磁式跟踪器的主要优点是传感器可自由移动;磁场可以穿透发射源和接收器之间的物体,不存在遮挡现象;可实现6自由度跟踪;接收器体积小重量轻,对被测物体的运动影响较小;在一定条件下分辨率很高。其缺点是在产生电磁场的物体(如显示器等)、金属物体和一些办公设备的周围会受到干涉,导致其自身的磁场受到干扰。另外,电磁跟踪设备的工作容积通常也比较小。

2. 声学式

声学式位置跟踪器通常由声波发生器、接收器和处理单元构成。声波发生器不间断地发射一个短促的超声波,接收器内部一般至少由三个超声波探头构成,这样就可以测量信号从发生器到达各个超声波探头的时间差,从而计算出接收器的位置和方向。

声学式位置跟踪器的优点主要是可以解决人体的遮挡问题,而且成本相对较低。但是它缺点很多,包括易受到其他声音源的干扰、运动数据的捕捉延迟较大、精度不高且实时性较差等。而且由于气压、温度和湿度对声音传播速度造成的影响,所以必须在算法中进行补偿。

3. 光学式

光学式位置跟踪器需要给被跟踪对象佩戴上标记点,然后跟踪标记点以实现人体的运动捕捉。其原理是如果空间中的一个标记点可以被多台摄像机所见,而且这些摄像机都是已经标定的,那么通过摄像机所拍摄的图像就可以计算出这个标记点在三维空间中的位置。

光学式运动捕捉系统是目前商业中使用最广泛的运动捕捉技术,其优点是不需要使用电缆连接被跟踪对象和设备、对象运动受限制小、采样率高,可以满足大多数高速运动测量的需要。但是这种方法使用过程中经常会发生标记点的混淆和遮挡,导致运动信息的缺失或者计算错误,所以需要后期的人工干预比较多。由于需要对标记点进行识别、跟踪和空间坐标的计算,致使后期处理的工作量非常大。光学式运动捕捉系统价格比较昂贵,安装定位较烦琐,对场地的灯光和反射情况也有一定的要求。

4. 机械式

机械式跟踪器由一个串联或并行的运动结构组成,该运动结构由多个带有传感器的关节连接在一起的连杆构成。虚拟现实仿真中使用的第一个跟踪器是支撑 Sutherland 研究的机械臂,该机械臂固定在天花板上,可以跟踪用户头部的相对运动。每个连杆的维数是已知的,可供存储在计算机中的直接运动学计算机模型使用,并可根据实时读取的跟踪器关节传感器的数据,确定机械式跟踪器的某个端点相对其他端点的位置和方位。

与其他跟踪技术相比,机械式跟踪器非常简单且易于使用,在跟踪器工作范围内,它的精度相对稳定,仅取决于关节传感器的分辨率。在所有类型的跟踪器中,机械式跟踪器的抖动比较小,延迟比较低。与光学式跟踪器不同,机械式跟踪器与被跟踪对象之间没有视觉阻挡问题。机械式跟踪器最明显的缺点是其工作范围有限,如果连杆过长,其重量和惯性会随之增加;另一个缺点是由于跟踪器机械臂自身运动的牵制,用户运动的自由度受限。

参考文献

1. Gutiérrez M A, Vexo F, Thalmann D. Stepping into Virtual Reality. London: Springer, 2008

2. Craig A, Sherman W R, Will J D. Developing Virtual Reality Applications: Foundations of Effective Design. Burlington, MA: Morgan Kaufmann, 2009

(王涌天　刘越)

wenben fenlei

文本分类(text classification)　将文本集中的每个文本按预先给定的分类体系自动分到某个或某几个类别中去的算法和实现技术。文本分类可看作监督学习(参见**非监督学习**)的一种形式,与文本分类相对比,文本聚类则通常被看作无(非)监督学习的一种形式,预先不给出类别体系和类别差异,主要通过相似度计算将文本聚集成多个类或簇,使得同一类或簇中的文本内容具有较高的相似度,而不同类或簇中的文本内容差异较大。

文本分类过程可以分为手工分类和自动分类。前者的实例如 yahoo 的网页分类系统,由专家先定义分类系统,然后人工分类网页。这种方法需耗费大量人力,已经很少采用。自动文本分类算法大致又可分为两类:**知识工程**方法和**机器学习**方法。知识工程方法指的是由专家为每个类别定义一些规则,这些规则反映了这个类别的特点,机器自动把符合规则的文档划分到相应类别中。20世纪90年代之后,机器学习方法成为主流,能够达到知识工程方

法的准确度,并大量减少了人工参与,取得了成果并逐渐取代了知识工程方法。

基于机器学习方法的文本分类通常分三个步骤:①文本表示,最常用的方法是向量空间模型,即把文本集表示成词—文档矩阵,矩阵中每个元素代表了一个词在相应文档中的权重。选取一些词来代表文本,这个过程称为特征选择。常见的特征选择方法利用文档频率、信息增益、互信息、期望交叉熵等。为了减少分类过程中的计算量,经常需要进行降维处理,比如潜语义分析(latent semantic indexing,LSI)等方法。②分类器构建,选择或设计构建分类器的方法。没有一种通用方法可以适用所有情形,不同的方法有各自的优缺点和适用条件,要依据问题特色来选择一个分类器。常见的分类方法包括:Rocchio 方法、朴素贝叶斯(naive Bayes)方法、*K* 近邻(*K*-nearest neighbor, KNN)方法、支持向量机(support vector machine,SVM)方法等。③分类结果评估,对分类结果进行评测。常用的评估标准由信息检索领域而来,包括召回率、准确率、F1 值等。

基于机器学习方法的文本分类技术较少考虑文本的语义信息,将语义分析和概念网络等方法与机器学习方法相结合会取得更好的分类效果。

参考文献

1. Yang Yiming. An evaluation of statistical approaches to text categorization. Journal of Information Retrieval, 1999, 1(1/2): 67-88

2. Mamming C D, Raghavan P, Schutze H. 信息检索导论. 王斌译. 北京:人民邮电出版社,2010

(李素建)

wenben neirong chacuo

文本内容查错(text content error check) 计算机根据语言本身包含的信息,利用**自然语言处理**技术对电子文本进行自动分析,发现、标示其中各种错误的技术和过程。

英文文本内容查错的研究始于 20 世纪 60 年代,主要包括英文单词拼写检查和语法检查,在“非词错误”和“真词错误”两个层次上进行。非词错误也称单词错误,是指文本中被词边界(通常是“空格”)分隔出的字符串不是词典中的词条,其查错和纠错方法主要有误拼词典法、词形距离法、最小编辑距离法、骨架键法等技术。真词错误也称上下文错误,指某个字符串虽然是词典中的词,但它与上下文搭配不当,其查错和纠错方法主要有基于上下文同现与搭配特征法、相邻字词间的接续关系分析法以及基于语言学知识与规则的方法等。

中文文本查错研究始于 20 世纪 90 年代初期。由于中文和英文在文本结构、构词形态、字符录入计算机的方式以及字符集规模大小等方面的不同,中、英文文本内容查错的方法也有所不同。中文文本中不会出现非字错误,只可能出现别字、错词或外语单词拼写错误以及由此引起的句法语义错误。一个字或词之所以被认为是别字或错词,是由于它与所处的上下文环境不相适应,这种不相适应可以从字词级、句法级和语义语用级来考察。

(1) 字词级错误　主要由别字、多字、漏字、易位、外文单词误拼等引起的错误及标点符号错误,也可将其分为非词错误和真词错误两类。非词错误是指错误造成的字串不是词典中的词,真词错误则是指错误造成的字串是一个可以在词典中查到的词,但与其上下文环境不相适应。

(2) 句法级错误　指某些错误虽然破坏了原词结构,但由于能与前后字成词,或单字本身成词,因而并不造成词法错误,但却破坏了句子的整体结构,造成语法错误。比如词性搭配错误、关联词搭配错误或句型错误等。

(3) 语义语用级错误　指语法正确,但语义不合常规的现象。一般情况是由于别字、多字、漏字、易位等错误导致一个词变成了另一个词性相同的词。还有一种情况是由于作者用词不当或字词错误,引起词与词之间或短语与短语之间的语义搭配错误。

中文文本内容查错目前主要是字词级的查错处理,而句法、语义只是作为补充手段。常用方法有:①利用文本上下文的字、词和词性等局部语言特征,构建查错模型;②构建词二元转移概率矩阵,对相邻词间的接续关系进行分析,设计查错算法;③利用句法规则和语言学知识,设计查错算法;④统计与规则相结合的方法等。

参考文献

江铭虎. 自然语言处理. 北京:高等教育出版社,2006

(张仰森)

wenben wajue

文本挖掘(text mining)　对文本数据的挖掘(参见**数据挖掘**)。是从大量文本数据中抽取出事先未知的、可理解的、最终可用的知识的过程,同时运用这些知识更好地组织信息以便将来参考。从实

际应用角度出发,文本挖掘是要从非结构化文本信息中获取用户感兴趣或者有用的模式。

信息时代的发展产生了海量的各式各样的数据,其中大量数据以文本形式存在,如新闻文档、研究论文、书籍、数字图书馆、电子邮件、Web 页面等等。由于电子形式的文本信息飞速扩展,文本挖掘已成为当今信息领域的研究热点,也是实际应用中的重要技术之一。文本挖掘是一个多学科交叉的领域,涵盖了信息技术、**自然语言处理**、认知语言学、信息检索、**模式识别**、统计学、数据可视化、数据库技术、**机器学习**以及**数据挖掘**等技术。

文本挖掘过程可大致分为三个环节:文本预处理、模式发现、结果评估与展现。文本预处理主要是根据任务选取相关的文本并将其转化为挖掘工具可处理的中间形式。模式发现是指利用机器学习、数据挖掘、模式识别等方法提取出面向特定应用目标的知识或模式。结果评估与展现是利用已定义好的评估指标对获取的模式进行评价,如果符合要求则将该模式存储以便用户使用,否则返回前面某个环节重新调整和改进,进行新一轮的挖掘。文本数据本身的特点如高维、海量、复杂语义等,要求文本挖掘应具备:高效性,即运行的时间较短和空间复杂度较低;健壮性,以处理富含大量噪声和不规则结构的文本。

自 1995 年 Feldman 提出文本挖掘概念以来,文本挖掘作为人们处理信息爆炸式增长与有效利用之间矛盾的重要工具,在世界范围日益受到重视。目前国际上许多机构都在进行文本挖掘技术的研究,并取得一定的成绩,特别是在拉丁语系国家发展迅速。研究主要围绕文本特征抽取、特征约简、表示模型构建、分类、聚类、主题分析、趋势分析、情感分析等,已经形成一套较成熟的理论体系和技术手段。同时,文本挖掘在多个领域得到了广泛的应用,如客户关系管理、自动邮件回复、垃圾邮件过滤、自动简历评审、搜索引擎、话题识别与追踪、在线新闻实时监控、专利数据分析、文献挖掘、问卷调查分析等。

就我国文本挖掘研究的现状而言,由于中文文本的特殊性,中文文本挖掘还处在相对初级的阶段。现已有研究者将汉语本身的特点融合到文本挖掘中,研究中文文本的构成特点与特征提取机制,充分做好中文文本预处理的工作(选择、清洗、分词、特征提取等),为中文文本挖掘提供强有力的支持。

参考文献

1. Feldman R, Sanger J. The text mining handbook. Cambridge University Press, 2007

2. Srivastava A, Sahami M. Text mining: Classification, clustering, and applications. Boca Raton, FL: CRC Press, 2009

3. Bilisoly R. Practical text mining with Perl. New York: John Wiley & Sons, 2008

(景丽萍　黄厚宽)

wenben zidong chuli

文本自动处理(automatic text processing)

通过建立形式化的模型,用计算机对自然语言文本进行自动化分析和处理的一系列方法和技术。文本自动处理,也可称为文档自动处理,所涉及的研究范围非常广泛,从汉字输入法、编辑排版到信息检索和自动文摘等,只要是借助计算机和文本打交道,均可统称为文本自动处理技术。早期的文本自动处理专注于形式层面,将文本看作是一种与其他数据类似的信息,相关的处理技术通常称为文本信息加工,例如汉字编码法、中文排版等。从某种程度上说,文本信息加工只是从表层处理文本,忽视了文本内容所携带的语言特性。

随着信息需求的增长和语言学的发展,研究者逐渐加强了对文本内容的处理。这种处理开始关注文本是语言信息承载者这一特点,从而把注意力聚焦在语言理解问题上。一方面人们逐渐对自动理解语言能力的局限性有了更多的了解,另一方面在这个过程中也积累了很多副产品,拓宽了对计算机和语言文本结合的认识,从理解转变为处理。由此,与文本自动处理密切相关的术语还包括“**自然语言处理**”“**自然语言理解**”“人类语言技术”等。

文本自动处理技术按照所处理的文本对象的不同,可分为形态分析(着重于如何用计算机自动分析语言中的基本组成成分——词)、句法分析(主要以自然语言句子为研究对象,分析句子的结构和语义)、篇章分析(主要以比句子更大的单位——篇章为分析对象开展的研究)等。研究方法通常可以区分为规则方法和统计方法两大类,在文本自动处理技术的发展历史中出现过的各种具体方法,基本上都可以归入上述其中的一大类或两大类方法的融合。

现代科技的高速发展使得海量文本信息涌现出来,只有有效的文本处理技术才可以把人们从中解脱出来。在这种需求的驱动下,文本自动处理衍生了各种应用任务,其目标是研发出以数字化文本为

处理对象的准确高效的信息组织和存取系统。Gerard Salton 在 20 世纪 70 年代,为 SMART 检索系统建立索引时,采用了文本自动处理技术对文本内容进行分析和标引,通过实验对比显示出优于基于受控词表的传统标引技术。除了**自动标引**和信息检索(参见**信息检索方法**)技术,面对海量文本信息的典型应用系统还包括**信息提取**、**文本分类**、文本聚类、**自动文摘**等。这些应用系统通常采用比较成熟的词法分析技术,进一步的智能化还有赖于自然语言处理技术的深入研究。

参考文献

1. Salton G. The SMART Retrieval System—experiments in automatic document processing. Upper Saddle River, NJ: Prentice-Hall, Inc., 1971

2. 俞士汶. 计算语言学概论. 北京:商务印书馆,2003

(李素建)

wendang yuyan

文档语言(documentation language)　用于书写计算机软件文档的语言。计算机软件文档是计算机软件开发、维护和使用过程的档案资料和对软件本身的阐明性资料。

根据文档的作用和性质,常见的计算机软件文档可分为:

(1) 对软件本身的描述,如软件基准手册;

(2) 对软件使用方式的描述,如用户指南、程序人员指南、联机帮助等;

(3) 软件开发过程的记录和中间产品与结果,如需求定义、**功能规约**、**设计规约**、测试计划、测试报告、维护与修改记录等。

书写前两种软件文档的语言主要是自然语言,辅以图表等。这些文档要求语言简练准确,通俗易懂,还经常以一定的结构储存在计算机系统中,如以超文本的形式,以便用户联机检索。

对于书写软件开发过程中产生的中间结果的语言,则通常根据中间结果的特性使用不同的语言。如需求定义常采用 E-R 图、**状态转移图**、**数据流图**等,加上自然语言描述(参见**需求定义语言**)。书写软件功能规约的语言既可是形式化的**功能规约语言**,如 **Z 语言**、**OBJ 语言**等,也可是自然语言。软件的设计规约可使用特定的软件设计语言书写(参见**设计规约**),也可采用软件结构图加上自然语言来描述。良好的测试计划、维护与修改记录等应当是按照标准的格式使用自然语言书写。软件测试报告通常以自然语言填写的表格的形式出现。

(朱鸿　金凌紫)

wenfa

文法(grammar)　语言结构的一种有限描述。给定任意的有限字母表 Σ,Σ^* 表示由 Σ 中的字母组成的所有符号串(包括空串)的集合。Σ^* 的每个子集都是 Σ 上的一个语言。形式语言主要研究这种界限明确的语言。N. Chomsky 于 1959 年提出的生成文法是形式语言的一种描述手段。形式上,生成文法是一个四元组

$$G = (\Sigma, V, S, P)$$

其中 V 是有限的变量集合,又称为非终结符号表;Σ 是有限的字母集合,又称为终结符号表,$V \cap \Sigma = \varnothing$;$S$ 是开始符号,$S \in V$;P 是有限的形如 $\alpha \to \beta$ 的生成式集合,$\alpha \in (V\cup\Sigma)^* V (V\cup\Sigma)^*$,$\beta \in (V\cup\Sigma)^*$。生成式又称为规则。在不对文法 G 的生成式增加限制时,称文法 G 为无限制文法,或者短语结构文法,或者 0 型文法。

推导　由文法生成语言的句型的过程。给定文法 $G = (\Sigma, V, S, P)$ 和 $\alpha', \beta' \in (V\cup\Sigma)^*$,如果 $\alpha' = \beta'$ 或者存在 $\alpha_1, \alpha_2, \alpha, \beta \in (V\cup\Sigma)^*$,使得 $\alpha' = \alpha_1\alpha\alpha_2$,$\beta' = \alpha_1\beta\alpha_2$ 且 $(\alpha\to\beta) \in P$,则称在文法 G 中 α' 直接推导为 β',记为 $\alpha' \underset{G}{\Rightarrow} \beta'$。用 $\overset{*}{\underset{G}{\Rightarrow}}$ 表示 $\underset{G}{\Rightarrow}$ 的自反传递闭包。如果 $S \overset{*}{\underset{G}{\Rightarrow}} \alpha$,$\alpha \in (V\cup\Sigma)^*$,则称 α 为文法 G 生成的句型。文法 G 生成的所有的句型的集合

$$S(G) = \left\{\alpha \in (V\cup\Sigma)^* \mid S \overset{*}{\underset{G}{\Rightarrow}} \alpha\right\}$$

例如短语结构文法 $G_1 = (\{a, b\}, \{S, A, B, C\}, S, P_1)$,其中 $P_1 = \{S\to ASa, S\to BSb, S\to C, AC\to aC, BC\to bC, Aa\to aA, Ba\to aB, Ab\to bA, Bb\to bB, C\to\lambda\}$,则有推导

$$S \underset{G_1}{\Rightarrow} ASa \underset{G_1}{\Rightarrow} ABSba \underset{G_1}{\Rightarrow} ABCba \underset{G_1}{\Rightarrow} AbCba \underset{G_1}{\Rightarrow} bACba \underset{G_1}{\Rightarrow} baCba \underset{G_1}{\Rightarrow} baba$$

这里 $S, ASa, ABSba, ABCba, AbCba, bACba, baCba$ 和 $baba$ 都是 G_1 生成的句型。

短语结构语言　短语结构文法生成的所有句子的集合。如果文法 $G = (\Sigma, V, S, P)$,$\alpha \in S(G)$ 且 $\alpha \in \Sigma^*$,则 α 是文法 G 生成的句子。文法 G 生成的语言

$$L(G) = S(G) \cap \Sigma^* = \{w \in \Sigma^* \mid S \overset{*}{\underset{G}{\Rightarrow}} w\}$$

0 型文法生成的语言称为 0 型语言或者短语结构语言。

例如在上面给出的文法 G_1 中，$S \overset{*}{\underset{G_1}{\Rightarrow}} baba$，故 $baba \in \Sigma^*$ 是 $L(G_1)$ 的一个句子。实际上，文法 G_1 生成的短语结构语言 $L(G_1) = \{uu | u \in \{a, b\}^*\}$。

参考文献

1. Hopcroft J E, Ullman J D. 自动机理论、语言和计算导引. 徐美瑞，译. 北京：科学出版社，1986

2. Chomsky N. On certain formal properties of grammars. Information and Control, 1959, 2 (2): 137-167

（郭清泉）

wenhua chengxu sheji

文化程序设计（literate programming） 一种程序设计方法，将程序设计视为著述活动，目标是使程序员写非常容易理解的软件。它提供软件支持，允许程序员完全按照自己思路的本来逻辑顺序来写程序，并且同时产生可执行代码及具有高可读性的程序说明文件。

这种设计方法主张，应该把程序看成具有网状结构：一个较复杂的软件片段是由若干个较简单的片段和它们之间的简单关系所构成的，每一个这样的片段自然又可以由更简单的片段来构成。程序员的任务是以一种最有利于人们理解的次序来表述这些片段和其间关系，而不是固定地按照某种确定次序，例如自顶向下或者自底向上。这样，无论是软件的作者还是读者，都得到可理解的软件。

文化程序设计首先由 D. E. Knuth 于 20 世纪 80 年代所倡导，并在 1983 年研制出第一个文化程序设计系统 WEB。WEB 语言所提供的设施允许人以“意识流”的次序来表达程序，程序员可以把大程序看成一张网，以一种从心理上看是正确的次序来探索它。WEB 的语言包含了两种成分，一种是供描述算法过程的 **PASCAL 语言**，另一种是描述程序文档排版要求的 TEX 排版语言。

WEB 系统由两个子系统组成，一个子系统从 WEB 语言程序中自动地抽出描述算法过程的部分，并且加工成 PASCAL 编译程序所能接受的形式，然后据此得到可以在计算机上执行的代码。另一个子系统则是把 WEB 语言程序加工为 TEX 系统所能接受的形式，并据此得到具有高度可读性的程序说明文件。文化程序设计的两种组成语言是可以更换的。例如曾经有 **C 语言**和 TROFF 的组合，C 和 TEX 的组合（CWEB），C 和中西文排版语言 SP 的组合（CDS）。被采用过的其他程序语言还包括 Modula-2，FORTRAN，Ada 等。

有几个软件系统，即 WEB 和 CWEB 本身，TEX 和 METAFONT，已经被 D. E. Knuth 等人用 WEB（或者 CWEB）写出，并且公开出版。它们也都已经被列为自由软件，任何需要的人均可以从互连网络上获取。

下文的图 1 是 quick sort 算法的一个完整程序文档（为紧凑起见，消除了空白，页面用水平线隔离）。它是图 2 中的 CWEB 源程序经系统加工而得到的。

QSORT

---

§1. QSORT

QUICK SORT FUNCTION FOR ARRAY 1

1. Quick sort function for array. This function sorts or partially sorts an array into increasing order. Every element of the array has a key which is comparable, and sorting is according to the value of the keys. After this sorting process, the elements are rearranged and the keys of the elements are in increasing order.

2. This function uses “Quick sort” algorithm. This algorithm uses the divide and conquer technique. To begin each iteration an element is selected from the file and the file is then split into two subfiles, those elements whose keys are smaller than the selected one and those elements whose keys are larger. In this way, the selected element is placed in its proper final location between the two resulting subfiles. This production is repeated recursively on the two subfiles and so on. Quick sort is a very popular sorting algorithm; although its worst case is $O(n * n)$, its average performance is excellent. This worst case occurs when the file is in order already. Any portion of the file that is nearly in order will significantly deteriorate Quick sort's efficiency.

3. Action definitons are related to what the element type is defined. **KEY**(*elemt*) is the key

of the element. *Assign* (*elemt*1, *elemt*2) sets the value of *elemt*1 to be the one of *elemt*2.

#define KEY(*elemt*) *elemt*

#define*Assign* (*elemt*1, *elemt*2) *elemt*1 ⇐ *elemt*2;

4. The program structure is here

< Type definitions 5 >

< Function 6 >

5. **ArrayEntry** is the type name of the elements of the array to be sorted.

ArrayToSort is the type name of the array to be sorted.

< Type definitions 5 > ≡

typedef int Array Entry;

typedef ArrayEntry ArrayToSort[ ];

This code is used in section 4.

6. Now is the Quick sort function. *r* is the array to be sorted, *lo* and *up* are the lower and upper bound, respectively, of array r.

< Function 6 > ≡

sort (**ArrayTosort** *r*, **int** *lo*, **int** *up*)

{

< Local variables 7 >

< Iteration loop 8 >

}

This code is used in section 4.

7. We need a temporary unit *temp* to hold the selected element, also two variables *i* and *j* to point to currently processed elements.

< Local variables 7 > =

int *i*, *j*;

ArrayEntry *temp*;

This code is used in section 6.

§ 2. QUICK SORT FUNCTTION FOR ARRAY QSORT § 8

8.

< Iteration loop 8 > =

while (*up* > *lo*)

{

i⇐*lo*;

j⇐*up*;

Assign(*temp*, *r*[*lo*]);

(Split file in two, elements whose keys are smaller than *temp* are in *r*[*lo*. . *i* − 1], and ones with larger keys are in *r*[*i* + 1. . *up*]9)

Assign(**r**[*i*], *temp*); / * i is the correct location for temp * /

sort(*r*, *lo*, *i* − 1); / * Sort recursively * /

lo⇐*i* + 1;

}

This code is used in section 6.

9.

< Split file in two, elements whose key are smaller than *temp* are in *r*[*lo*. . *i* − 1], and ones with larger keys are in *r*[*i* + 1. . *up*]9 > ≡

While(*i* < *j*)

{

While(KEY(*r*[*j*]) > KEY(*temp*))

j⇐*j* − 1;

r[*i*]⇐*r*[*j*];

While(*i* < *j* ∧ KEY(*r*[*i*]) ≤ KEY(*temp*))

i⇐*i* + 1;

r[*j*]⇐*r*[*i*];

}

This code is used in section 8.

§ 10 QSORT INDEX 3

10. Index.

4 NAMES OF THE MODULES QSORT § 10

< Function 6 > Used in section 4.
< Iteration loop 8 > Used in section 6.
< Local variables 7 > Used in section 6.
< Split file in two, elements whose keys are smaller than *temp* are in *r*[*lo*. . *i* − 1] , and with larger keys are in *r*[*i* + 1. . *up*]9 > Used in section 8.
< Type definitions 5 > Used in section 4

图 1 Quick sort 算法的程序文档

@* Quick sort function for array.

This function sorts or partially sorts an array into increasing order. Every element of the array has a key which is comparable, and sorting is according to the value of the keys. After this sorting process, the elements are rearranged and the key of the elements are in increasing order.

@ This function uses "Quick sort" algorithem. This algorithm uses the divide and conquer technique. To begin each iteration an element is selected from the file and the file is then split into two subfiles, those elements whose keys are smaller than the selected one and those elements whose keys are larger. In this way, the selected element is placed in its proper final location between the two resulting subfiles. This production is repeated recursively on the two subfiles and so on.

Quick sort is a very popular sorting algorithem; although its worst case is | O(n * n) | ,its average performance is excellent. This worst case occurs when the file is in order already. Any portion of the file that is nearly in order will significantly deteriorate Quick sort's efficiency.

@ Action definitions are related to what the element type is defined.
| KEY (elemt) | is the key of the element.
| Assign (elemt1 , elemt2) | sets the value of | elemt1 | to be the one of | elemt2 | .

@ C

#define KEY (elemt) elemt
define Assign (elemt1, elemt2) elemt1 = elemt2;

@ The program structure is here

@ C

@ < Type definitions @ > @ ;
@ < Function @ > @ ;

@ | ArrayEntry | is the type name of the elemts of the array to be sorted.
| ArrayToSort | is the type name of the array to be sorted.

@ < Type. . . @ > =
typedef int ArrayEntry;
typedef ArrayEntry ArrayToSort[ ];
@ Now is the Quick sort function.
| r | is the array to be sorted, | lo | and | up | are the lower and upper bound, respectively, of array | r | .

@ < Fun. . . @ > =
sort(ArrayToSort r, int lo, int up)
{
@ < Local variables @ > @ ;
@ < Iteration loop @ > @ ;
}
@ We need a temporary unit | temp | to hold the selected element, also two variables | i | and | j | to point to currently processed elements.
@ < Local. . . @ > =
int i,j;
ArrayEntry temp;

@ @ < Iter. . . @ > =
while (up > lo)
{i = lo;
j = up;
Assign (temp,r[lo]);
@ < split file in two, ele-

```
ments whose keys are smaller than | temp |
are in | r[lo..i-1] |,and ones with larger
keys are
  in | r[i+1..up] | @ >
              Assign (r[i],temp);
/ * |i| is the correct location for |temp| * /
              sort (r, lo, i - 1);  /*
              sort recursively * /
              lo = i + 1;
            }
@ @ < split...@ > =
  while (i<j)
     {while (KEY(r[j]) > KEY (temp))
              j = j - 1;
      r[1] = r[j];
      while (i < j && KEY (r[i]) < =
                KEY(temp))
                i = i + 1;
      r[j] = r[i];
     }
@ * Index.
```

图 2　Quick sort 算法的 CWEB 源程序

参考文献

1. Knuth D E. Literate programming. The Computer Journal, 1984,27(2):97-111

2. Thimbleby H. Experiences of ' Literate Programming' using cweb (a variant of Knuth' s WEB). The Computer Journal, 1986,29(3):201-211

3. Knuth D E,Levy S. The CWEB system of structured documentation. Version 3.0. New York: Addison-Wesley,1994

（董韫美　陈海明）

wenjian

文件(file)　有组织的数据的集合。

早期,用户按物理地址存取存储媒体上的信息,使用不便、效率很低。引入文件概念后,用户不再需要了解文件存放的物理位置和物理结构,可实现"按名存取",由文件管理程序根据用户给出的文件名自动地完成数据传输操作。把数据组织成文件加以管理是计算机数据管理的重大进展,其主要优点是:使用方便,安全可靠,便于共享。

从用户使用的角度看,文件可以分成两类(参见**文件管理程序**)。

在许多操作系统中,都把 I/O 设备看作是一个"文件",称设备文件,这样用户无须考虑保存其文件的设备差异,用统一的观点去处理驻留在各种存储媒体上的信息,给使用带来极大方便。

可以各种方式对文件进行分类。按用途可分成:系统文件、用户文件和库文件;按保护级别可分成:只读文件、读写文件、可执行文件和不保护文件;按存放时限可分成:临时文件、永久文件和档案文件;按设备类型可分为:磁盘文件、磁带文件、软磁盘文件;按信息流向可分成:输入文件、输出文件和输入输出文件。

参考文献

孙钟秀,等.操作系统教程.4 版.北京:高等教育出版社,2008

（郑宇华）

wenjian chuansong

文件传送(file transfer)　一台计算机在另一台计算机中按一定格式复制文件的技术。互相传送文件的两台计算机可以是不同类型的,如微型计算机与大型计算机;也可以是安装不同操作系统的,如 DOS 与 UNIX。

要实现文件传送,必须在计算机上有相应的进程来启动,很多文件传送协议都允许文件的双向传送。例如,客户可以把文件发送给服务器,也可以向服务器请求文件并接收文件。为进行这样的传送,调用客户程序的用户必须给出自己的标识符,并且获得许可。本地许可由本地操作系统解决(如根据登录的标识符和口令在登录时获得许可)。访问远程文件的用户则必须得到对客户机授权的服务器的许可之后才能进行文件传送。

传送文件需要遵循一定的通信协议,以确保端对端传送的正确性。常用的文件传送协议有 XON/XOFF、Kermit、XMODEM、YMODEM、FTP、TFTP 和 FTAM 等,其中以 FTP 最为流行。

文件传送协议 FTP

文件传送服务是计算机网络中使用最广泛的应用之一。在互联网中,文件传送服务采用文件传送协议(FTP),因此,通常用 FTP 表示文件传送服务。通过 FTP,用户可与远程主机连接,直接将远程主机文件系统中的文件全部拷入本地文件系统,也可将本地文件全部拷进远地文件系统。在计算机网络中有成千上万个文件服务器供用户获取资源,包括公用程序、原始程序代码、研究报告、技术文档以及各

类论文等。

FTP 是基于客户服务器模型而设计的，与其他客户服务器模型不同之处在于 FTP 客户与服务器之间要建立双重连接，一是控制连接，另一是数据连接。建立双重连接是因为 FTP 是交互式会话系统，客户每调用一次 FTP，便与服务器建立一次会话，该会话通过控制连接来维持，直至退出 FTP 为止。在一次 FTP 会话中，需建立一个控制连接和若干数据连接。控制连接负责传送控制信息，尤其是客户命令（如文件传送命令等）。利用这种命令，客户可以向服务器提出多次请求。客户每提出一个请求，服务器即与客户建立一个数据连接，进行实际的数据（如文件）传送。一旦数据传送结束，数据连接相继撤销，但控制连接依然存在，客户可以继续发出命令，直到客户键入 close 命令才撤销控制连接，再输入 quit 命令，便退出 FTP 会话。

FTP 的访问控制有两类：一类是严格的 FTP 访问控制，另一类是非严格的，前者要求客户给出文件所在的主机上的合法账号（包括注册名和口令），才能访问主机；后者由支持匿名 FTP 服务器提供，便于广大客户获取主机中的公开文件，客户对这类服务器只要输入账号为 anonymous 和本地服务器所提示的口令即可。

通过计算机程序也可调用 FTP，这意味着文件传送能够自动进行。例如，可以设计一个应用程序在每天指定的时刻对某些文件进行检查，并且传送修改过的文件。运行这样的程序可以实现多种工作的自动化。

参考文献

1. 曹东启，等. 计算机网络软件基础. 北京：人民邮电出版社，1982

2. 梁振军，等. 计算机互联网技术与 TCP/IP 协议. 北京：海洋出版社，1991

3. 胡道元. 计算机局域网. 3 版. 北京：清华大学出版社，2002

（李学农）

wenjian guanli chengxu

文件管理程序（file manager） 操作系统中用于管理和存取数据文件的程序。它采用统一、标准的方法管理在辅助存储器上用户和系统文件数据的存储、检索、更新、共享和保护，并为用户提供一整套操作和使用方法。实现文件管理的程序模块集合称为文件系统。

文件逻辑结构指用户概念中文件数据的排列方法和组织关系。有如下两类：

（1）流式结构 文件内的数据是依次的一串字节集合。这种文件称流式文件。

（2）记录式结构 文件内是顺序的若干个逻辑记录的集合，逻辑记录是文件中按数据在逻辑上的独立含义来划分的信息单位。这种文件称记录式文件。

文件物理结构指文件数据在存储空间中的存放方法和组织关系，有两类方法：

（1）计算法 设计一个映射算法，通过对记录键的计算转换成逻辑记录的物理地址，从而存取记录。散列文件、顺序文件均属此类。

（2）指针法 用指针来表达逻辑记录之间的关系，从而，找到逻辑记录的物理地址。索引文件、串联文件、倒排文件均属此类。

文件系统给每个文件建立唯一的管理数据结构，称文件控制块 FCB，其中，包含：文件名、文件属性、文件地址等信息。为了加快文件的查找，通常把 FCB 集中起来进行管理，便组成文件目录。文件目录是用来管理文件系统结构的系统文件，它是实现“按名存取”文件的主要手段和工具，文件管理的基本功能之一就是文件目录的建立、检索和维护。文件目录结构可分成：一级目录结构、二级目录结构、树形目录结构。查找目录可用方法有：顺序查找、二分查找、分级查找和散列查找。

文件共享指一个文件可以让规定的某些用户共同使用。文件共享主要有两种方式：连接共享和符号链接共享。

文件保护和保密与文件的共享是互为依存的。文件保护指防止文件拥有者误用或授权者破坏文件；文件保密指不经文件拥有者授权，任何其他用户不得使用文件。两者均涉及用户对文件的访问权限。以下方法可规定使用权限：存取控制表、访问控制列表、文件使用权限。文件保密措施有：隐蔽文件目录、口令、密码。

文件存放在大容量辅存空间中，辅存空间可用以下方法管理：空闲区表、位示图、空闲块链、成组空闲块链等。

文件系统提供一整套操作和使用手段，供用户使用文件。提供的文件类系统调用有：建立、删除、打开、关闭、读、写、控制等；提供的文件类键盘操作命令有：查看目录、改变目录、列目录、建立目录、删除目录、文件改名、文件复制、文件显示、改变文件属性等。

参考文献

孙钟秀,等. 操作系统教程. 4 版. 北京: 高等教育出版社,2008 (费翔林)

wenyu zhuanhuan

文语转换(text to speech, TTS) 把文字自动转换成言语(语音)的技术。文语转换(TTS)是言语合成技术(参见**言语合成方法**)的延伸和扩展。TTS 系统中除了语音合成模块外,还包括文本分析、韵律生成模块,如图 1 所示。

文本分析是文语转换系统(TTS)的前端。它对输入文本进行分析理解,为后端语音合成器提供必要的信息,比如读音、停顿、韵律等信息。不同的合成后端需要的信息也各不相同。对于简单系统,可能文本分析只需要提供读音信息就够了;对于高自然度的任意文本的合成器,文本分析要给出更详尽的语言学或语音学信息,使得合成器合成的语音拥有更多的可调节余地。结合自然语言处理和人工智能的研究成果,在充分"理解"文本的基础上,它的输出信息尽可能做到有正确的读音,有轻重缓急的标记,甚至包含不同的感情风格等。应该说,理想的文本分析器同时就是一个理想的**自然语言理解**程序。

文本分析一般包括文档结构分析、文本规范化、语法分析、韵律分析、字音转换五个部分。其中,文本规范化和字音转换是两个重要的过程。文档结构分析是对输入文本的结构进行划分,如果输入文本是某种带标记的格式(比如 SSML 语言标记),文档结构分析也担负着对标记的解释工作。文本规范化将文本中可能存在的约定俗成的书写方式,比如数字、日期、符号等,转换为标准的书写形式(othographic)。语法分析对文本进行语法分析,确定单词、短语和句子的构成。对于汉语来说,由于汉语的书写文本不像英语等语言具有单词的分隔符,所以分词是**汉语信息处理**中一项比较独特的工作。韵律分析根据语法分析的结果以及其他相关信息确定句子的韵律结构,重读部分,这些信息是建立韵律模型必需的。字音转换(grapheme to phoneme conversion)确定每个单词或字符的读音。输入文本依次通过上述这些模块,每个模块会根据前面模块的输出信息,添加一些新的信息到输出流中。

韵律生成是指采用各种方法建立韵律模型,并在言语合成时产生韵律参数。韵律首先是一个听感知觉的概念,能帮助听者更好地理解话音所携带的信息。韵律在感知上表现为语音的音高、速度和音量随时间的变化,在声学参数上表现为基频、音段时长和能量随时间的变化。人们使用语调、节奏和重音等方式来表达说话者的意向和情感,这些韵律特征是自然语流的重要组成部分。要实现韵律模拟,需解决韵律规则、韵律描述、计算模型和修改算法等问题。这要借助于语音学、语言学、心理学、信号处理等学科的成果,研究韵律变化的特点,抽取韵律规则,找出韵律与声学参数的映射关系,给出定量的数学描述,建立计算模型,设计韵律修改算法。韵律建模是指计算言语中的韵律特性,建立韵律模型,以产生韵律参数。通常韵律模型分为基于规则的模型和

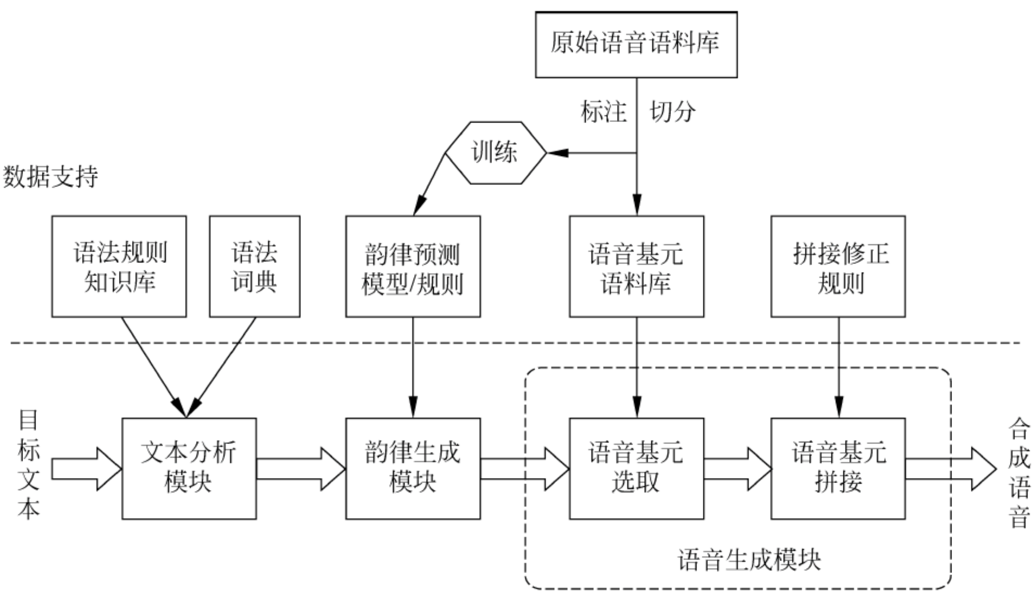


图 1 文语转换的处理过程

基于大规模语料库的数据驱动模型两类。

目前,合成语音的质量还欠理想,其根本问题是不能对自然语流中韵律进行有效的模拟。TTS 系统要想取得高质量的声音,必须具备韵律处理和模拟功能。语调、节奏和重音这些韵律特征是通过超音段特征——音高、音长、音强及频率分布的变化表现出来的。因此,这些超音段特征的修改成为韵律合成的基础。超音段特征的修改可通过多种方法实现,如修改基频模式、共振峰模式等。

参考文献

1. 蔡莲红,黄德智,蔡锐. 现代语音技术基础与应用. 北京:清华大学出版社, 2003

2. Huang X, Acero A, Hon HW. Spoken language processing: a guide to theory, algorithm and system development. Prentice Hall, 2001

(蔡莲红　杨鸿武)

wenli hecheng

纹理合成(texture synthesis)　产生物体表面细节的图形生成过程和技术。物体的表面细节称为纹理。纹理是表达物体质感的重要特性,通过纹理的生成可以使图形更具有真实感。通常有两类纹理合成的方法,一类是基于过程的纹理合成方法(procedural texture synthesis),另一类是基于样图的纹理合成方法(example-based texture synthesis)。

基于过程的纹理合成方法使用特定的函数来生成纹理,通常用来合成自然界中的自然现象,如大理石花纹、木纹、天空云彩等。经典的基于过程的纹理合成方法有基于柏林噪声的三维纹理、基于细胞状噪声函数的细胞状纹理和基于反应扩散过程的自组织纹理。基于过程的纹理合成方法的优点是计算速度快,无须消耗大量纹理内存,可以在绘制的时候实时生成纹理;其缺点是没有通用的计算模型,一般针对特定的纹理需要设计专门的合成函数,设计过程需要手工调试模型参数,调试过程不直观,需要反复调整参数才能生成想要的纹理。

基于样图的纹理合成方法根据输入的小幅纹理样图自动生成与原图视觉相似的任意大小的纹理,且合成结果没有明显的重复感。基于样图的纹理合成方法可以分为参数化的合成方法和非参数化的合成方法。参数化纹理合成方法基于人眼视觉的参数模型描述纹理,通过改变这些参数合成新的纹理。这类方法仅通过少量参数表示纹理,表示紧凑且易于进行纹理编辑。参数化纹理合成方法对随机性强的纹理合成效果较好,对于结构性较强的纹理合成效果较差。非参数化纹理合成方法采用马尔可夫随机场描述纹理,要求输出纹理图像中的每一个像素的邻域与纹理样图中至少一个邻域相似,通过在样图中查找当前像素的相似邻域进行合成。非参数化纹理合成方法可以大致分为基于像素的纹理合成、基于块的纹理合成和基于优化的纹理合成方法。其中基于优化的合成方法合成结果质量最优,基于像素的纹理合成灵活性高,可以提供像素级别上的编辑操作。在提高纹理合成速度方面,基于**图形处理器**的实时纹理合成算法大大加快了纹理合成的速度并使之应用于实时绘制。

比较上述两种纹理合成算法,过程纹理合成算法相比基于样图的纹理合成方法通用性较差,而基于样图的纹理合成算法相比过程纹理需要消耗更多的纹理内存、较难编辑且生成的纹理分辨率有限。但是近年来针对基于样图的纹理合成算法中的这些问题提出了很多解决方法,这些方法的提出使得基于样图的纹理合成算法更加实用。

参考文献

1. Wei L Y, Lefebvre S, Kwatra V, Turk G. State of the art in example-based texture synthesis. Eurographics 2009, State of the Art Report, EG-STAR, 2009

2. Turk G. Texture synthesis on surfaces. Proceedings of SIGGRAPH 2001, 347-354 (鲍虎军　华炜)

wenli yingshe

纹理映射(texture mapping)　为了表示物体表面细节,将一个纹理函数映射到物体表面上,并且在绘制该物体表面时,将纹理函数值作为参数来调节表面上各点的光亮度的技术和过程。采用纹理映射的益处在于可以使用较少的处理代价来表现物体表面细节,特别是当纹理函数为二、三维栅格数据时,便于使用图形硬件完成。通常,纹理映射至少包含从纹理空间到物体空间(纹理空间是指纹理函数存在的空间,物体空间是指物体表面存在的空间)的映射及从物体空间到屏幕空间(屏幕空间是指光栅化设备的逻辑坐标系)的映射。

大多数情况下,纹理函数是以二维图像形式描述的。在此情况下,纹理函数从纹理空间映射到三维物体表面的过程,直观上讲,就是将二维图像“贴”到三维物体表面上。这里首先需要解决的问题是物体表面参数化,即物体表面上任意点(x,y,z)

都可以表示成$(x,y,z)=\boldsymbol{v}(u,v)$，其中$\boldsymbol{v}$为一一映射，$(u,v)$为纹理坐标。在此基础上，再建立$(u,v)$坐标到图像位置$(i,j)$的关系，即$(u,v)=\boldsymbol{p}(i,j)$，其中$\boldsymbol{p}$为一一映射，通常称为纹理变换。这样，物体表面上任意一点(x,y,z)上的纹理值为图像上位置为$(i,j)=\boldsymbol{p}^{-1}(\boldsymbol{v}^{-1}(x,y,z))$的像素颜色所确定。在实践中，纹理映射的困难之处在于物体表面参数化，这体现在两个方面：①如何对任意拓扑结构任意几何形状物体表面构造如上所述的映射$\boldsymbol{v}$；②若要纹理在物体表面上不发生扭曲变形，应使物体表面的任意微小面元在通过上述映射后，在图像空间保持形状不变。然而，构造这样的映射$\boldsymbol{v}$和映射$\boldsymbol{p}$是相当困难的。

从物体空间到屏幕空间的映射通常发生在绘制过程中。当屏幕空间中某一像素被物体表面上某一面元所填充时，该像素上的纹理值应为该面元上的纹理均值。而在绝大多数的情况下，严格求出一个面元上的纹理均值在计算量上是不可接受的，因此，人们提出了多种近似计算面元上纹理均值的方法，如最近距离法、线性插值法、MipMap 法等。

不同的纹理映射技术主要表现在纹理函数及纹理空间到物体空间的映射方式上的不同。按纹理函数的维数来分类，有一维纹理、二维纹理、体纹理、动态纹理等；按照纹理函数的内容来分类，有颜色纹理、凹凸纹理、法向纹理、位移纹理等；按建立纹理空间到物体空间映射的方法来分类，有平面投影法、柱形投影法、球形投影法、立法体投影法、表面参数化法等。

参考文献

1. 彭群生，鲍虎军，金小刚. 计算机真实感图形的算法基础. 北京：科学出版社，1999

2. OpenGL Architecture Review Board. Shreiner D, Woo M, Neider J, Davis T. OpenGL® Programming Guide: The Official Guide to Learning OpenGL, Version 2.1 (6th Edition), 2007

3. [美]Rogers D F 著. 计算机图形学的算法基础(原书第 2 版). 石教英，彭群生，等译. 北京：机械工业出版社，2002

（鲍虎军　华炜）

wenda xitong

问答系统(question-answering system)　允许用户以自然语言问句的形式提供自己的需求，系统根据用户的输入，返回能以准确、简洁的答案回答用户问题的新型信息检索系统。

20 世纪 90 年代，随着信息技术、尤其是互联网的发展，问答系统的研究和开发转向了基于大规模文档集。TREC(Text Retrieval Conference)于 1999 年开始了问答技术的评测。2000 年 10 月，ACL(Association of Computational Linguistics)以“开放域问答系统”为专题，专门召开讨论会进行相关技术的讨论。从此以后，对开放域问答技术的研究就成了信息检索和自然语言处理技术领域一个重要的研究热点，TREC 的问答评测成为最受关注的评测项目之一。TREC 问答评测在评估系统返回的结果是否正确时，一方面要判断返回的答案是否正确，另一方面要判断从中提取该答案的文档内容是否支持该答案成立，即要求文档是和问题相关的文档。对于简单事实型问题，文章中同一个答案的表述可以有各种不同形式。因此，在构造问答系统的评测集时，标准答案的构造很难做到覆盖文档集中所有的答案。对于某些正确的答案，也有可能不能被列到标准测试集中。因此，类似 TREC 的各种问答评测，对系统性能的评判也只能是相对评估。对于定义型问题，由于答案是在不同侧面的定义描述，类似于文档文摘和机器翻译结果的评判，对这些描述很难进行简单的正确或者错误的判断。另外，不同的用户，对定义的期望也是不同的。因此相对于事实型问题来说，定义型问题更难评测。如何设计对定义型和复杂性问题的评测方法与评测指标，是评测组织者需要不断研究的问题。

对问答系统涉及的应用领域进行分类，可分为限定域问答系统和开放域问答系统。限定域问答系统是指系统所能处理的问题只能限定于某个领域或者某个内容范围，由于系统要解决的问题限定于某个领域或者范围，因此如果把系统所需要的全部领域知识都按照统一的方式表示成内部的结构化格式，则相对来说回答问题时就较易于产生答案。而开放域问答系统不同于限定域问答系统，这类系统可回答的问题不应当限定于某个特定领域。在回答开放领域的问题时，需要一定的常识或者世界知识，以及语义词典。

对支持问答系统产生答案的文档库、**知识库**，以及实现技术分类，可分为自然语言的数据库问答系统、对话式问答系统、阅读理解系统、基于常用问题集的问答系统、基于知识库的问答系统，以及基于大规模文档集的问答系统。

(1) 自然语言数据库问答系统大多是限定域的，数据库的设计均限定于某个特定领域，因此系统能处理的问句或查询要求涉及的知识领域限定在数

据库所收集的内容范围内。

（2）对话式问答系统区别于其他类型问答系统的一个重要特征，是能以对话形式和使用者交流，问题和相应答案的描述可以在一个上下文环境中。

（3）阅读理解测试是针对特定文档的问答任务。在该前提下，每个问题都针对一个特定文档，而答案也要求在该文档中。特定文档的问答和一般问答相比较，需要面对更大的挑战，因为在一般情况下，答案只在文档中出现一次；而一般问答所用的文档集中，一个答案可能会在多个文档中出现，因此也就有更多找到答案的机会。在现代阅读理解系统中，有些系统也会返回实际答案，而不是包含最可能包含答案的句子。

（4）基于常问问题集（frequently-asked question，FAQ）的问答系统是在已有的"问题—答案"对的集合中找到与用户提问相匹配的问题，并将其对应的答案直接返回给用户。通常问答系统都需要进行复杂的问题分析和答案生成。基于 FAQ 的问答系统，只需专注于用户输入的新问题和 FAQ 库中保存的问题之间的相似度比较。

（5）基于知识库的问答系统强调的是所存储知识的权威性和系统的推理能力。这类系统的优点是回答准确，可以进行一定的推理计算，缺点是需要建立大规模知识库，消耗大量人力物力。

（6）基于大规模文档集的问答系统通过对用户所提问题进行分析，根据问题从大规模文档集中查找到与之相关的文本，然后从这些文本中抽取出答案。

英文的问答系统已经获得了长足发展，产生了一批已有广泛商业应用的系统，如 Ask、AnswerBus、START 等。尤其是，2011 年 2 月，由 IBM 公司和美国得克萨斯大学历时四年联合研制的超级电脑"沃森"（Watson）在美国最受欢迎的智力竞猜电视节目《危险边缘》中击败该节目历史上两位最成功的选手，成为《危险边缘》节目新的王者。"沃森"就是一个超级问答系统，其中存储了海量数据，而且拥有一套逻辑推理程序，可以推理出它认为最正确的答案。"沃森"的成功激发了人们挑战机器问答系统极限的热情，也促使人们对于将问答系统应用于医学、法律、工程等各个领域增强了信心。

参考文献

1. Hirschman L, Gaizauskas R. Natural language question answering: the view from here. Natural Language Engineering, 2001, 7(4): 275-300

2. Verberne S, Boves L, Oostdijk N, Coppen P. Discourse-based answering of why-questions. Traitement Automatique des Langues Volume 47, Special Issue on Computational Approaches to Discourse and Document Processing (2007): 21-41

3. Riezler S, Vasserman A, et al. Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL 2007)[C]. Prague, Czech Republic: Association for Computational Linguistics, 2007: 464-471

（刘挺　邵艳秋）

wuhan yajie

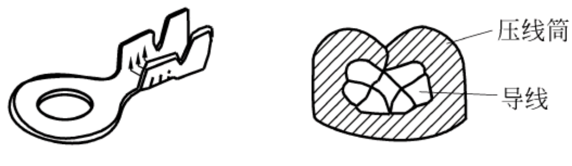
无焊压接（solderless crimp connection） 采用压接工具或设备，将导线插入可塑性金属的压线筒中，在压线筒的压接部位施加压力，使压线筒和导线在压接部位产生塑性变形，将金属表面的氧化层破坏，形成清洁无氧化层的接点，以达到稳定可靠的电气连接的技术。

早在 1882 年就有人提出压接的专利，但由于当时没有专用的端子和工具，未能推广应用，第二次世界大战期间才得到发展。现在压接端子有两种结构形式。一种是筒状银焊封口型，如图 1 所示；另一种是开口型，如图 2 所示。



(a) 外形图　　(b) 压接后剖面示意图

图 1　筒状银焊封口型压接端子



(a) 外形图　　(b) 压接后剖面示意图

图 2　开口型压接端子

开口型压接片便于做成连锁型端子，可以卷在圆盘中，便于压接自动化，提高了压接的生产效率，每小时可压上千到几万个连接点，远高于其他各种连接工艺。

压接工艺简单，效率高，最大的优点是导线连接时不用加热，不需使用助焊剂，连接后的导线不变

硬,不腐蚀,已大量地应用在各种连线和连接器中。如美国 AMP 公司能提供 54 000 余种自动、半自动、手动压接工具。压接工艺已被广泛地应用在数据处理设备、户内外供电设备、通信设备、家用电器、汽车、船舶以及各种军用电子设备的连线和连接器中。

20 世纪 80 年代开始,计算机采用大规模或超大规模集成电路后,各单元互连线大量减少,故机器中电气连接大量采用压接来代替绕接和焊接。

(顾本斗)

wuxian tu

无限图(infinite graph) 顶点集或边集为无限集的图。一般地,如果没有特别说明,图论中所讨论的图都是有限图。如果一个图中的所有顶点都只有有限的度(degree),那么这个图是局部有限(locally finite)图。一个顶点的度是指与该顶点相连的总边数,顶点 v 的度记作 $d(v)$。

有如下形式的无限图(V,E)叫做射线(ray):顶点集合 $V=\{x_0,x_1,x_2,\cdots\}$,边集合 $E=\{x_0x_1,x_1x_2,x_2x_3,\cdots\}$;有如下形式的无限图$(V,E)$叫做双射线(double ray):$V=\{\cdots,x_{-1},x_0,x_1,\cdots\}$,$E=\{\cdots,x_{-1}x_0,x_0x_1,x_1x_2,\cdots\}$,其中 x_n 互不相同。因此,本质上只存在一条射线和一条双射线,即所有射线和双射线都是同构的。双射线是一种独特的无限 2-正则连通图。一条射线或双射线的局部射线(subray)叫做该射线或双射线的尾(tail)。一条射线有无数多条尾,但其中任意两条尾的不同之处仅在于其有限的初始部分。

一条射线 R 与无数多个初始顶点在 R 上且互不相交的有限路径(path)的合集叫做梳(comb);这些路径的最后的顶点叫做这个梳的齿(tooth);射线 R 叫做梳的脊(spine),如图 1 所示。脊上的一个单独的点也可以被当作一个齿。

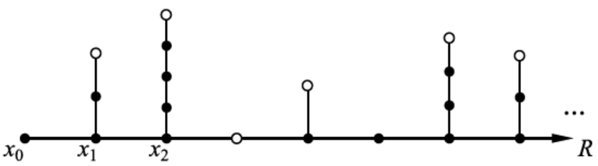


图 1 由白色的齿和脊 $R=x_0x_1\cdots$构成的梳

无限图中一个特有的概念是末梢(end)。无限图的末梢与边的端点(ends, endpoints, 或 endvertices)不是一个概念。为方便起见,下文中的图 G 均指无限图。图 G 的一个末梢是图 G 中射线的一个等价类。如果两条射线对于每一个有限集 $S\subseteq V(G)$,都在 $G-S$ 的一个相同的分支(component)上有一条尾,那么可认为这两条射线等价。图 G 的一个极大连通子图称为图 G 的一个分支(component)。当且仅当两条射线等价时,这两条射线可被无限多个不相交的路径连接。图 G 的末梢的集合记作 $\Omega(G)$。$G=(V,E,\Omega)$表示图 G 含有顶点集 V、边集 E 和末梢集 Ω。

如图 2 所示的双向无限梯形图,此图中的每条射线要么在图的左边,要么在图的右边的任意远处有顶点,但不可能同时在图的左边和右边的任意远处有顶点。因此,此图的射线有两个等价类,即此图有两个末梢(如图 2 中左右两边的两个孤立点所示)。

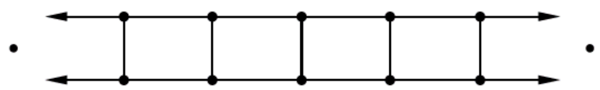


图 2 双向梯形图有两个末梢

上述例子说明,一个图的末梢可以被认为是其射线在无限远处的汇聚点。一个末梢所含的不相交射线(disjoint ray)的最大数目为此末梢的顶点度(vertex-degree),其所含的边不相交射线(edge-disjoint ray)的最大数目为其边度(edge-degree)。一个末梢必在 $N\cup\{\infty\}$ 中有一个顶点度和一个边度。有无限顶点度的末梢叫粗末梢(thick end);有有限顶点度的末梢叫细末梢(thin end)。

无限图有如下性质:①无限连通图必含有一个有无限度的顶点或含有一条射线;②令 U 是一个连通图 G 的顶点的无限集,那么 G 必含有一个梳及其所有齿在 U 中或含有一个无限星形的分图及其所有叶子在 U 中。

无限图的概念和性质可以用到人工智能的搜索问题求解中,当搜索的状态空间具有潜无限性,或者状态空间特别巨大时,应用无限图的理论和方法会显得比较方便。有些问题,比如网络的动态演化问题,采用无限图作为其理论描述就可以方便地建立起相应的数学模型,有利于进行深入的研究。

参考文献

1. Deo N. Graph theory with applications to engineering and computer science. Upper Saddle River, NJ: Prentice-Hall Inc., 1974

2. Diestel R. Graph theory. New York: Springer-Verlag, 2005

(江志斌)

wuxian mesh wang

无线 mesh 网(wireless mesh network, WMN)

一种多跳、自组织的宽带无线网络,由 mesh routers(路由器)和 Mesh clients(客户端)组成,通常采用分级网络结构:mesh 路由器构成骨干网络,负责数据的中继,并通过网关节点与 Internet、蜂窝通信网等其他网络互联;mesh 客户端通过 Mesh 路由器接入骨干网,实现 mesh 客户端之间的通信,或经过骨干网访问其他网络。

Mesh 客户端可以是手提电脑、手机、PDA 等装有无线网卡、天线的客户设备,客户端之间可以互联构成小型 Adhoc 网络,在用户设备间提供点到点服务。

无线 mesh 网的结构如图 1 所示。

与传统的 WLAN 相比,无线 mesh 网络具有以下优势:

(1) 快速部署和易于安装　Mesh 的设计目标是将有线设备和有线 AP 的数量降至最低,因此安装方便,节省成本。

(2) 非视距传输(NLOS)　由于采用多跳无线连接,信号能够自动选择最佳路径不断从一个用户跳转到另一个用户,并最终到达无直接视距的目标用户。无线 mesh 网络能够非视距传输的特性大大扩展了无线宽带的应用领域和覆盖范围。

(3) 健壮性　Mesh 网络比单跳网络更加健壮,因为它不依赖于某一个单一节点的性能。由于每个节点都有一条或几条传送数据的路径。如果最近的节点出现故障或者受到干扰,数据包将自动选路到备用路径继续进行传输,整个网络的运行不会受到影响。

(4) 结构灵活　在无线 mesh 网络中,每个设备都有多个传输路径可用,网络可以根据每个节点的通信负载情况动态地分配通信路由,从而有效地避免了节点的通信拥塞。

(5) 高带宽　在 mesh 网络中,一个节点不仅能传送和接收信息,还能充当路由器对其附近节点转发信息,随着更多节点的相互连接和可能的路径数量的增加,网络的总带宽也大大增加。

多跳网络通常使用较低功率将数据传输到邻近的节点,节点之间的无线信号干扰也较小,网络的信道质量和信道利用效率较高,因而能够实现更高的网络容量。

目前无线 mesh 网络还没有统一的技术标准,设备之间的互操作性较差,网络中的通信延迟和安全问题也有待进一步改进。

当前无线 mesh 网络还存在几个需要解决的问题:

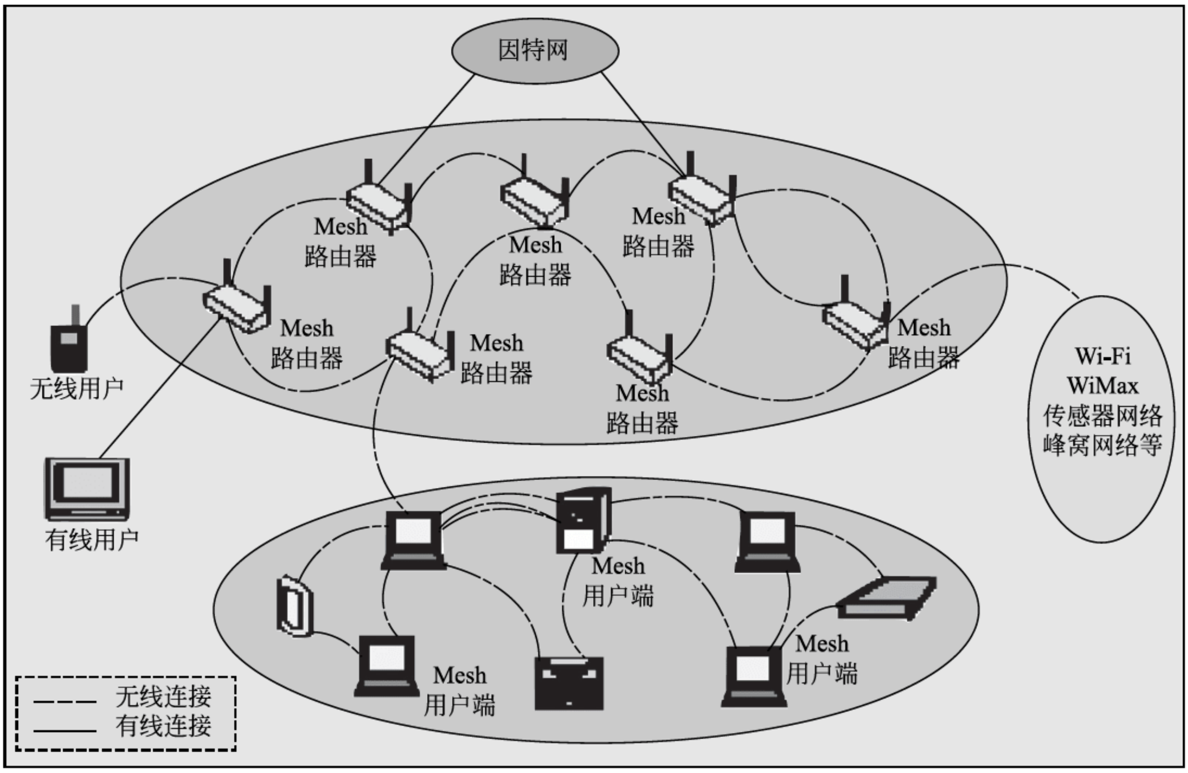


图 1　Mesh 网络混合结构图

（1）互操作性　无线 mesh 网络现在还没有一个统一的技术标准，这个问题目前是影响无线 mesh 技术推广使用最重要的原因。鉴于此，目前一些公司正在开发能够适应不同无线环境的可配置的无线网络设备，互操作性有望得到一定程度的解决。但要想彻底解决互操作性问题，最终还需要业界制定统一的无线 mesh 技术标准。

（2）通信延迟　在 mesh 网络中数据通过中间节点进行多跳转发，每一跳至少都会带来一些延迟，随着无线 mesh 网络规模的扩大，跳接越多，积累的总延迟就会越大。一些对通信延迟要求高的应用，如话音或流媒体应用等，可能面临无法接受的延迟过长的问题。目前解决这一问题主要是通过增加 mesh 节点以及合适的网络协议。随着多无线 mesh 节点技术的出现这一问题将得到最终解决。

（3）安全　与 WLAN 的单跳机制相比，无线 mesh 网络的多跳机制决定了用户通信要经过更多的节点。而数据通信经过的节点越多，安全问题就越变得不容忽视。正如 Internet 一样，无线 mesh 网络的安全是一个不容忽视的问题。

参考文献

1. 方旭明. 下一代无线因特网技术：无线 Mesh 网络. 北京：人民邮电出版社，2006

2. [加]霍辛（Hossain，E.），[英]梁（Leung，K. K.）. 无线 Mesh 网络架构与协议. 易燕，等译. 北京：机械工业出版社，2009

3. http://www.cww.net.cn/tech/html/2008/5/6/200856113758260l.htm

4. http://net.zdnet.com.cn/network_security_zone/2008/1024/1194584.shtml

5. http://baike.baidu.com/view/1215700.html

（程时端）

wuxian baozhen

无线保真（Wi-Fi）　将个人电脑、手持设备（如 PDA、手机）等终端以无线方式互相连接的技术，事实上它是一个高频无线电信号。无线保真是一个无线网路通信技术的品牌，由 Wi-Fi 联盟所持有。

（周正）

wuxian juyuwang

无线局域网（wireless local area network，WLAN）　一种利用无线射频技术，取代双绞铜线所构成的**局域网**络，传送距离通常只有几十米。WLAN 用户可以通过一个或几个固定的接入点（AP）连接到互联网，具有良好的便利性和移动性。

WLAN 标准由 IEEE 802.11 工作组制定，第一个版本发表于 1997 年，其中定义了物理层和介质访问接入控制层（MAC 层）。1999 年工业界成立了 **Wi-Fi 联盟**，致力解决符合 802.11 标准的产品的生产和设备兼容性问题。

802.11a 占用 5.3 GHz 自由频段，在 10 m 范围内其速率可高达 54 Mbit/s。

802.11b 占用 2.4 GHz 的自由频段，其速率理论上可以达到 11 Mbit/s。

802.11g 其实是一种混合标准，它既能适应传统的 802.11b 标准，在 2.4 GHz 频率下提供 11 Mbit/s 数据传输率，也符合 802.11a 标准在 5 GHz 频率下提供 54 Mbit/s 数据传输率。

802.11n 采用多输入多输出（MIMO）技术和正交频分复用（OFDM）技术，传输速率可以从 54 Mbit/s 增加至 108 Mbit/s 以上，其最高数据速率预计可达 320 Mbit/s。802.11ac 是 802.11n 的继承者，占用 5 GHz 频段，理论上，它能够提供最少 1 Gbps 多点通信带宽，或最少 500 Mbps 的点到点传输带宽。

组网结构

简单的家庭无线 LAN　在家庭无线局域网最通用和最便宜的例子，无线路由器可以提供广泛的功能，如防火墙、路由器、交换机和无线接入点。通过 NAT 转换，允许共享一个 ISP（Internet 服务提供商）的单一 IP 地址，也可以和另一个以太网交换机或集线器进行扩展。

中型无线局域网　中等规模的企业传统上使用的一种设计，简单地向所有需要无线覆盖的设施提供多个接入点。大多数这类无线局域网允许在接入点之间漫游，因为它们配置在相同的以太子网和 SSID 中。从管理的角度看，每个接入点以及连接到它的接口都被分开管理。从安全的角度来看，每个接入点必须能够处理其接入控制和认证，比如通过 RADIUS 服务器授权，因此也被称为胖 AP。

大型可交换无线局域网　简化的接入点通过几个中心化的无线控制器进行控制。这种情况下的接入点具有更简单的设计（也称瘦 AP），用来简化复杂的操作系统，而且更复杂的逻辑被嵌入在无线控制器中。接入点通常没有物理连接到无线控制器，它们利用隧道协议逻辑上通过无线控制器交换和路由。

无线局域网安全

在未加密的 Wi-Fi 网络连接设备上,可以监测和记录数据(包括个人资料)。有线等价保密(WEP)是最常见的无线加密标准,但即使在正确配置下,已被证明是很容易遭到攻击的。为了解决这个问题,制定了 Wi-Fi 保护访问(WPA 和 WPA2)加密,于 2003 年在设备中得到使用。

参考文献

1. IEEE 802.11 工作组. http://www.ieee802.org/11/

2. WiFi 联盟. http://wi-fi.org/ (金耀辉)

wuxian wangluo anquan

无线网络安全(wireless network security)

主要包括蜂窝移动通信网、无线局域网、近距离通信网等网络的安全。无线网络安全威胁主要来自于无线网络协议和系统的弱点,攻击者可以在无线信道非法窃听信令和数据信息,非授权访问和处理网络资源,干扰、滥用网络服务,不同的安全威胁会给网络带来不同程度的破坏。

蜂窝移动通信网络安全

蜂窝移动通信网经历了第一代、第二代、第三代、第四代蜂窝移动通信网。

第一代模拟蜂窝移动通信网几乎没有采取安全措施,移动终端把其电子序列号和网络分配的移动台识别号以明文方式传送至网络,若二者相符,即可实现用户的接入,这种方式容易造成大量的克隆手机。

第二代数字蜂窝移动通信网(2G)主要有基于时分多址的全球移动通信(GSM)系统和基于码分多址(CDMA)的系统,二者使用的安全机制相似:①提供用户身份的保密措施,在用户初次接入网络的时候国际移动用户标识才被发送,仅在无线信道上发送移动用户相应的临时移动用户标识;②网络对用户进行单向的身份认证和授权,保证合法用户能够接入网络;③在移动终端和基站设备之间实现用户数据和信令数据的加密保护。

第三代数字蜂窝移动通信网(3G)在 2G 的基础上进行了改进,继承了 2G 系统安全的优点,例如用户身份保护机制,同时针对 3G 系统的新特性,定义了更加完善的安全特征:①用户和网络之间进行双向的身份认证和鉴权,保证合法用户能够接入网络,同时保证用户能够接入合法网络;②在移动终端和基站设备之间实现用户数据和信令数据的加密保护、信令数据的完整性保护;③移动终端和网络之间提供密钥协商机制。

第四代数字蜂窝移动通信网(4G)安全沿用 3G 网络的用户身份保护机制、双向身份认证和鉴权、密钥协商机制,根据 4G 系统的扁平化网络架构,定义了新的安全特性:①4G 网络安全包括两个层次,接入层(AS)安全和非接入层(NAS)安全,使得无线接口和核心网络安全相互独立,从而提高整个系统的安全性;②接入层安全实现移动终端与基站设备之间信令数据的加密和完整性保护、用户数据的加密保护;非接入层(NAS)安全实现移动终端与移动管理实体(MME)之间信令数据的加密和完整性保护;③采用分层的密钥体系架构,由密钥 K 派生出较多层次的密钥,分别实现各层的保密性和完整性保护。

无线局域网络安全

无线局域网使用的安全通信协议大致有三种,分别是 WEP(wired equivalent privacy)协议、WPA(Wi-Fi Protected Access)协议和 WAPI(WLAN Authentication and Privacy Infrastructure)协议。

WEP 是 802.11b 采用的安全标准,用于提供一种加密机制保护数据链路层的安全。WEP 采用 RC4 算法实现对称加密,通过预置在接入点(AP)和无线网卡间共享密钥。WEP 要求传输程序创建一个特定于数据包的初始化向量,将其与预置密钥相组合,生成用于数据包加密的加密密钥。接收程序接收此初始化向量,并将其与本地预置密钥相结合,恢复出加密密钥。WEP 存在较明显的安全缺陷:密钥长度太短(只有 40 比特);不支持自动更换密钥,所有密钥必须手动重设,会导致相同密钥的长期重复使用;初始化向量明文传递,允许在 5 个小时内重复使用,无法加强密钥强度。而且,WEP 采用的 RC4 算法被证明是存在漏洞的。

作为 WEP 的升级,WPA 分为 WPA 和 WPA2 两个版本,是 802.11i 的组成部分。不同于 WEP,WPA 同时提供加密和认证,保证了数据链路层的安全,并保证了只有授权用户才可以访问无线局域网络。WPA 采用 TKIP 协议(Temporal Key Integrity Protocol)作为加密协议,提供密钥重置机制,增强了密钥的有效长度。认证采取两种方法,一种采用 802.11x 协议方式,一种采用预置密钥 PSK 方式。

WAPI 是我国自主研发的无线局域网安全标准,是一种认证和私密性保护协议,其作用类似于 802.11b 中的 WEP,能提供更加完善的安全保护。WAPI 采用非对称(椭圆曲线密码)和对称密码体制

(分组密码)相结合的方法实现安全保护,实现了设备的身份认证、链路验证、访问控制和用户信息在无线传输状态下的加密保护。WAPI 除实现移动终端和 AP 之间的相互认证之外,还可以实现移动网络对移动终端及 AP 的认证。

近距离通信网络安全

蓝牙　蓝牙网络采用跳频扩频技术和低发射功率等常规安全技术,还采用内置的安全机制来保证无线传输的安全性。在蓝牙技术中定义了三种安全模式:①无安全要求。在这种模式下蓝牙设备屏蔽链路级的安全功能,适于非敏感信息的数据库访问。②强制业务级安全。这种模式提供业务级的安全机制,允许更多灵活的访问过程,蓝牙设备在信道建立后启动安全性过程,安全过程在较高层协议进行。③强制链路级安全。这种模式提供链路级的安全机制,蓝牙设备在信道建立前启动安全性过程,安全过程在较低层协议进行。

蓝牙网路定义了三种设备安全级别,可信任设备、不可信任设备和未知设备。定义了三种业务安全级别,需要授权与认证的业务、仅需认证的业务以及对所有设备开放的业务。

蓝牙网络用于确保安全传输的密钥有三种,链路密钥、加密密钥和个人识别码(PIN)。链路密钥用于两个蓝牙设备之间进行认证,加密密钥由链路密钥推算出来,确保数据包的安全,每次传输都会重新生成密钥。PIN 码用于设备之间互相识别。

蓝牙网络加密算法对通信双方的所有信息进行加密,由于密钥长度从 8 比特到 128 比特不等,信息交互双方必须通过协商确定密钥长度。

蓝牙网络的认证机制采用"询问—应答"式,首先认证服务器向提出请求的客户机发送一个需要认证的随机数,然后等待客户端回送结果,如果收到的结果与本地计算的结果相同,就表明认证成功。

Zigbee　ZigBee 提供了高可靠的安全服务,包括密钥建立、密钥传输、帧保护和设备管理等安全策略。ZigBee 安全体系结构包括三层安全机制,媒介存取控制层(MAC)、网络层(NWK)和应用层(APS)。MAC 层保证 MAC 命令帧、信标帧和响应帧的安全性;NWK 层提供用于构建不同网络拓扑结构的路由安全功能;APS 层提供建立和保持安全关系的服务。

ZigBee 为了实现安全性,定义了信任中心的角色。一个网络中只有一个信任中心,且被网络所有设备识别和信任。信任中心提供三种功能,信任管理、网络管理、配置管理。为了实现信任管理,设备需要接收信任中心使用非安全方式传输的初始主密钥或者网络密钥。为了实现网络管理,设备应接收初始的网络密钥,并且只能从信任中心获得网络密钥的更新。为了实现网络配置,设备需要从信任中心接收主密钥或链路密钥,以建立两个设备间的端对端安全链路。

参考文献

1. 任伟. 无线网络安全. 北京:电子工业出版社, 2011

2. 李晖, 牛少彰. 无线通信安全理论与技术. 北京:北京邮电出版社, 2011

(王志勤　袁琦)

wuxian yingyong xieyi

无线应用协议(wireless application protocol, WAP)　一个使移动用户的无线设备(例如移动电话)随时使用互联网的信息和服务的开放规范。用户可以借助无线手持设备,如掌上电脑、手机、呼机、双向广播、智能电话等,通过 WAP 获取互联网业务。

WAP 能够运行于各种无线网络之上,如 GSM、GPRS、CDMA 等。支持 WAP 技术的手机能浏览由无线注标语言(WML)描述的 Internet 内容。WML 是以 XML 为基础的标记语言,它支持文字和图片显示,内容组织上,一个页面为一个 Card,而一组 Card 则构成一个 Deck。当使用者向服务器提出浏览要求后,WML 会将整个 Deck 发送至客户端的浏览器,使用者就可以浏览 Deck 里面所有 Card 的内容,而不需要从网络上单独下载每个 Card。因此,通过 WAP 这种技术,就可以将 Internet 的大量信息及各种各样的业务引入到移动电话、PALM 等无线终端之中。WAP 目前主要使用 1.2 和 2.0 两个版本,后者是趋势,低端手机只能浏览 1.2 版本,目前主流的手机都已经支持 2.0 版本。

(张平)

wuxiang tu

无向图(undirected graph)　每条边均是无向边的图(参见**图论**)。设无向图 $G=\langle V,E,\psi\rangle$,对任意顶点 $v\in V$,称与 v 关联的边的数目为顶点 v 的度,记为 $d_G(v)$。G 的极大连通子图称为 G 的分支。所有顶点的度均为自然数 d 的无向图称为 d 度正则图。设 $n\in I_+$(正整数集合),如果 n 阶简单无向图 G 是 $n-1$ 度正则图,则称 G 为完全无向图,记为

K_n。设 n 阶无向图 G 是 n 阶完全无向图 K_n 的生成子图,则从 K_n 中删去 G 中的边,所得到的图称为 G 的补图,记为 $\overline{G}$。包含图中每条边恰好一次的路径,称为欧拉路径。包含图中每条边恰好一次的回路,称为欧拉回路。包含图中每个顶点恰好一次的路径,称为哈密顿路径。包含图中每个顶点恰好一次的回路,称为哈密顿回路。有欧拉回路的无向图称为欧拉图。有哈密顿回路的无向图称为哈密顿图。无向图的邻接矩阵和路径矩阵的定义与有向图一致(参见**有向图**)。设无向图 $G=\langle V,E,\psi\rangle$ 无自圈,$V=\{v_1,v_2,\cdots,v_n\}$ 且 $E=\{e_1,e_2,\cdots,e_m\}$,则 G 的关联矩阵 $\boldsymbol{A}(G)$ 是一个 $n\times m$ 矩阵(a_{ij}),其中,

$$a_{ij}=\begin{cases}1, & \text{若 } v_i \text{ 与 } e_j \text{ 关联}\\ 0, & \text{否则}\end{cases}$$

(张强)

wuyuan guangxian yonghu xianlu(xPON)

无源光纤用户线路(xPON) **光纤接入网**(optical access network,OAN)是采用光纤作为传输介质的接入网,泛指本地交换机或远端设备与用户之间采用光纤通信或部分采用光纤通信的系统。根据接入网室外传输设施中是否含有源设备,OAN 可划分为**无源光网络**(passive optical network,PON)和**有源光网络**(active optical network,AON)。PON 作为 OAN 的一种,它本身是一种多用户共享系统,即多个用户共享同一套设备,同一条光缆和同一个光分路器,所以成本低。与 AON 相比,其安装、开通和维护运营成本低,但系统可靠性、稳定性却很高,并且具有透明宽带传输能力。因此,PON 具有广阔的应用前景。

目前市场上的 PON 产品按采用的技术主要分为,EPON(以太网 PON)和 GPON(吉比特 PON)。

(1) GPON 是基于 ITU-TG. 984. x 标准的宽带无源光网络接入技术,由于 GPON 是在 APON 的基础上专门针对 APON 的缺点发展起来的,所以 GPON 保留了 APON 的许多优点,更高效、高速,提供从 622.080 Mb/s 到 2.4Gb/s 的可升级框架结构,支持上下行不对称速率,支持多业务,在下行方向采用 TDM 广播方式,在上行方向采用 TDMA 方式,可以灵活地组成树型、星型、总线型等拓扑结构。因此,GPON 成为目前最理想的宽带光接入网技术。

(2) EPON 是由 IEEE 802.3 工作组在 2000 年 11 月成立的"第一英里以太网(EFM)"研究小组提出的,采用 PON 拓扑结构实现以太网的接入的标准,2004 年 6 月,IEEE 802.3ah 被正式通过成为 EPON 国际标准。EPON 相对于现有类似技术的优势主要体现在:高带宽,提供高达 1Gb/s 的传输速率;低成本;易兼容,各个厂家生产的 EPON 网络设备易互联互通;具有强大的 QoS 支持能力。EPON 的网络结构可以支持光纤到户、光纤到楼等多种方式,承载的业务包括 IP 业务、时分复用业务和有线电视业务。

参考文献

1. 柯赓. 接入网技术与应用. 西安: 西安电子科技大学出版社, 2009

2. 雷维礼, 马立香, 等. 接入网技术. 北京: 清华大学出版社, 2006

3. 杨威. 宽带接入技术与实践. 北京: 人民邮电出版社, 2008

(程时端)

wuzhangai jisuanji jishu

无障碍计算机技术(accessible computer technologies) 为盲、聋、哑等残障人士能方便地操作计算机以克服信息交互障碍的技术。对于视力、听力退化的老年人,也可利用无障碍计算机系统改善生活质量。

一般来说,计算机给人们提供的信息主要是三类:图像、文字和声音。

听力没有损失的盲人,接收声音信息和正常人没有什么区别。而图像信息,包括静止图像和活动图像(电影、动画等),必须具有同步的声音播放,盲人才能获得相应的信息。早期计算机功能简单,没有声音、图形之类的多媒体信息。在此基础上开发盲人计算机就必须特制语音合成卡,开发文语转换系统,以及适合盲人使用的键盘输入方法。原先,盲人既不能阅读汉字,也不能书写汉字,他们只能使用盲文(六点不同组合的符号)。盲人计算机的出现,使他们有可能学习和使用汉字。所以说,盲人计算机为盲人打开了明眼文字的窗口。

聋哑人通常使用手语,经过艰苦的学习也能学会普通人使用的文字。不过,他们看电视时,如果没有字幕说明或同步手语演示,就没法理解视频播放的语言内容。现有的计算机语音识别技术可将语音识别成文字,让他们阅读。进一步,还可用专用软件将文字信息转化为手语,在屏幕上显示出来;再进一步,开发手语与普通文字之间的互译系统更能方便聋哑人与普通人之间的信息交流。

随着技术的进步,计算机功能愈来愈强大,能实

现更多更复杂的信息无障碍功能。现在,即便是最普通的个人计算机,在操作系统中就配有屏幕放大软件,供低视力人使用。如计算机系统配置语音识别、语音合成软件和文字识别、图形处理软件,就可以实现更好的视觉听觉功能,为盲聋哑人克服信息交互的障碍。

互联网的出现又给盲人计算机提出了新课题。由于互联网上的网页材料大都用 Jave 语言编写,充满了图形乃至动画和视频资料,要想滤除无用信息,使盲人上网取得所需资讯,就要编制特殊的读屏软件。从另一方面来讲,也要求网页的编写尽量做到无障碍化。现在已有许多信息无障碍网站,便于盲人上网。

其实,非残障的所谓正常人,也可能在特定场合下遇到各种信息交互障碍,例如,某人置身于国外,听不懂、看不懂、也不会说当地的语言文字,他就成为某种意义上又盲又聋又哑的信息交互障碍者。利用最先进的无障碍技术,包括文字识别、语音识别和**机器翻译**以及语音合成系统等技术手段,有可能解决(或减轻)这类信息交互的障碍问题。现在的数码相机和智能手机的结合,已有可能开发这类文字识别和翻译系统。有了这种无障碍设备,就能看懂、听懂、也能说任何一处的语言文字。这种技术对一般的视障、听障者更具实用意义。

由此可见,信息交互无障碍的概念是十分广泛的。“无障碍计算机技术”也超越了一般意义的计算机应用技术。具有**人工智能**的手机可能成为更广泛意义上的无障碍设备。

参考文献

信息无障碍专刊. 互联网天地. 2009, 11

(茅于杭　张红光)

Wu fangfa

吴方法(Wu method)　用计算机证明几何定理的一种方法,是我国著名数学家吴文俊于 20 世纪 70 年代末,将中国古代数学中的算法化思想应用于几何定理证明,独立发展的关于几何定理机器证明的一套理论及方法。1977 年,吴文俊给出了一类平面几何问题的机械化证明方法,并运用这种方法在机器上证明了一大批平面几何定理。这种几何定理的机械化证明方法,国际上称为“吴方法”。1984 年,周咸青发表的论文“Proving elementary geometry theorems using Wu's algorithm”以及 W. W. Bledsoe 等编辑的文集“Automated Theorem Proving: After 25 Years”使得“吴方法”在国际自动推理研究领域受到广泛关注。1997 年,吴文俊因这一突破性先驱工作而荣获自动推理领域的最高奖——Herbrand 奖。

定理机器证明能够以一种确定的机械方式,一步步给出对某一类定理证明过程的构造性步骤,并在有限步骤之后,或完成证明过程,或指出定理不成立。相对于定理的手工证明,机械化证明虽然缺少人类思维中的奇思妙想,但既能快速执行纷繁复杂的符号运算,又能轻而易举地保证证明过程的严密性,甚至也能得到一些原本未知的新定理。

一个几何定理包含着假设与结论,证明就是从假设到结论的判断过程。选取适当的坐标系统,用 $x_1, x_2, \cdots, x_n$ 等表示坐标,如果定理的假设可以写成

$$HS\begin{cases} h_1(x_1,\cdots,x_n) = 0 \\ \qquad\vdots \\ h_k(x_1,\cdots,x_n) = 0 \end{cases}$$

而结论可以写成

$$G \quad g(x_1,\cdots,x_n) = 0$$

其中 $h_1,\cdots,h_k,g$ 均为某一域上的多项式,则定理的证明就转化为:对于满足 HS 的任意一组 $(x_1^0,\cdots,x_n^0)$,判定是否有 $g(x_1^0,\cdots,x_n^0)=0$? 从几何上看,条件 HS 的每一组零点代表了一幅几何构图,定理的证明就是判定由 $(x_1^0,\cdots,x_n^0)$ 所确定的几何构图是否具有 G 这样的性质? 即要计算 $\mathrm{Zero}(h_1,\cdots,h_k)\subseteq\mathrm{Zero}(g)$ 是否成立,这里 $\mathrm{Zero}(f_1,\cdots,f_k)$ 表示多项式 $f_1,\cdots,f_k$ 的公共零点集。

我们仅粗略地看 $h_1,\cdots,h_k,g$ 都是 x 的一元多项式的情形,考察多项式组 $h_1,\cdots,h_k$ 与多项式 g 的零点集之间的关系。用 $h_k(x)$ 去除 $g(x)$ 得余式 $r_k(x)$,即 $g(x)=q_k(x)h_k(x)+r_k(x)$;用 $h_{k-1}(x)$ 去除 $r_k(x)$ 得余式 $r_{k-1}(x)$;…;用 $h_1(x)$ 去除 $r_2(x)$ 得余式 $r_1(x)$,则 $g(x)=\sum_{i=1}^{k}q_i(x)h_i(x)+r_1(x)$。若 $r_1(x)\equiv 0$,则有 $\mathrm{Zero}(h_1,\cdots,h_k)\subseteq\mathrm{Zero}(g)$。

对于一般的多元多项式,也有这种类似的“求余”运算、“余式”等概念,尽管一般情况下“余式”未必为零,但这种思想却是吴方法的出发点。吴方法中,通过对多项式集建立偏序关系、“求余”预算和约化,在有限步内将 $h_1,\cdots,h_k$ 化为一极简捷的“同解”多项式组 CS(这里的“同解”实际上是 $\mathrm{Zero}(h_1,\cdots,h_k)\subseteq\mathrm{Zero}(CS)$)$CS$ 称为 $h_1,\cdots,h_k$ 的特征列。从 $h_1,\cdots,h_k$ 到 CS 的机械化实现过程称为吴消元法。正如上述对一元多项式的分析那样,根据 $g(x_1,\cdots,x_n)=0$ 能否由方程组 $CS=0$ 推出,可以判定出定理

的正误。

机械化证明原理(吴方法)设几何定理的假设条件由多项式组 HS 表示,结论由多项式 G 表示。又 HS 的特征列为 CS: $C_1, \cdots, C_r$。如果 G 对 CS 的余式为零,则在非退化条件下,$G=0$ 可由 $CS=0$ 推出,即定理成立。

"吴方法"部分地基于代数几何,但与代数几何又有明显区别。代数几何中既有构造性证明又有存在性证明,以存在性证明为主,而且强调证明过程的精巧和直观,以便于被人理解和接受;而"吴方法"则是要给出可计算方法和构造性证明,更注重机械证明过程中计算的高效和严密。

"吴方法"已远远超出了**定理机器证明**的范畴,它作为多项式组的一种机械化处理方法,在国际上受到极大重视。经过吴文俊教授以及其他学者的多年努力,数学机械化这一研究领域已经被确立,研究成果也被广泛应用于符号计算、图形图像处理、计算机辅助设计、机器人等众多领域的科学研究和实际工程系统。

参考文献

吴文俊. 几何定理机器证明的基本原理(初等几何部分). 北京: 科学出版社,1984

(孙吉贵 欧阳丹彤 董旭初)

wuli sheji

物理设计(physical design) 将表示为逻辑门网络或晶体管网络的电路转换为由多边形组成的几何图形(即用于制造掩膜的电路版图)的一系列过程。它是集成电路设计的一个重要步骤。物理设计要完成两大类工作,一部分是实现工作,即把设计抽象层次高的电阻-晶体管逻辑(RTL)级或门级网表转换成可用于流片加工的版图。另一部分是验证工作,验证得到的布局布线后的实际版图是否满足电学性能要求、是否符合加工规则。

物理设计的步骤包括版图规划(floorplan)、布局(placement)、时钟树规划、电源网络规划、布线(routing)、信号完整性分析、可靠性分析、寄生参数提取、物理验证等。这些步骤之间有序传递设计信息和设计数据,形成芯片物理设计流程,如图 1 所示。

布图规划是按照一定的原则(比如更好地满足时序、最小面积或者更容易保证各电路模块之间的连接)将组成电路的各个模块摆放到二维平面上。这里的电路模块通常是有完整功能的独立模块,如存储器、乘法器、第三方**知识产权**(IP)核等。须在模块间留下足够的布线区域。

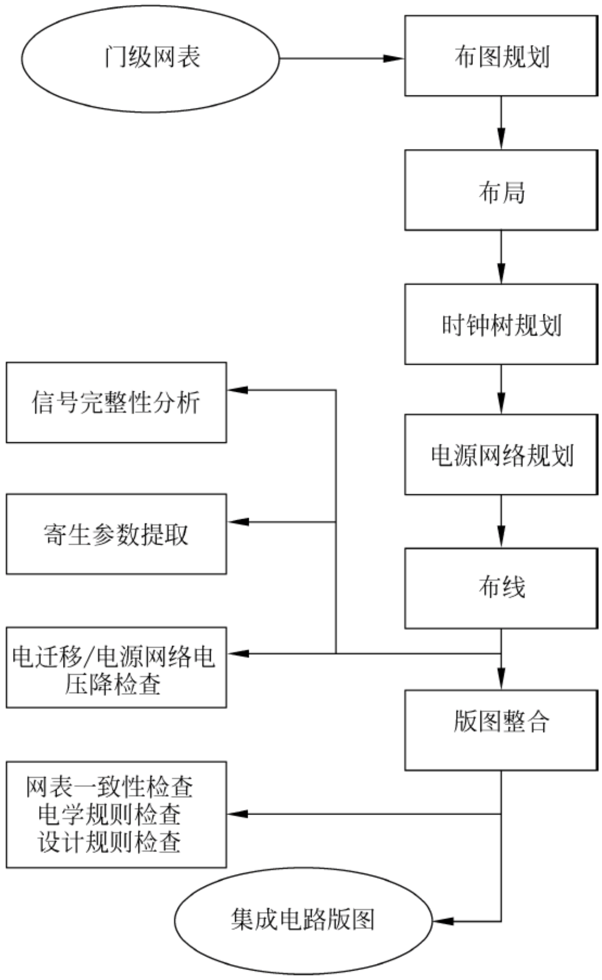


图 1 物理设计流程

布局是根据设计指标的要求,对各个电路模块内元器件进行具体的摆放和网表的优化。对于复杂的设计,可采用层次化的布局方法,即先将版图分割成多个区域,先在区域里布局,然后进行集成。

因为大部分芯片中的同步时钟都采用树形结构,时钟分布网络常被称为时钟树。时钟树规划也称时钟树综合,是通过加延迟单元等方式保证同一棵时钟树上各叶节点的时钟相位相同,时序偏差最小。

电源网络规划是对分布在全芯片各个位置的电路进行供电电源布局,以保证位于芯片不同位置的电路模块得到同样电位的供电。一方面,电源网络应尽可能均匀,保证在电源线上距离电源最近点和最远点的电压降处于允许的范围;另一方面,电源线要设计得足够宽以提供电路工作需要的电流,在预期的使用周期内不会因为电迁移引起电源线断开等

可靠性问题。

布线是把布局后的时钟树、电路模块以及电路模块内的元器件用符合加工设计规则的互连线几何图形连接,形成只包含标准单元和第三方 IP 核的连线框图以及互连线版图的全芯片版图。完成布线后只需要将标准单元、IP 核的版图整合到布线后的版图就形成完整的版图。布线是影响物理设计性能的关键步骤,有时序优先、布线布通率优先或者面积优先等不同的布线算法。

信号完整性分析是指对长互连线信号传输过程中受到的干扰,以及干扰引起的信号噪声是否会影响电路性能和功能的分析工作。在有影响的情况下,需要进行信号完整性修复,主要方法是加宽互连线的宽度和增加间距。

电迁移分析主要用于分析芯片电源网络的可靠性。电迁移是指用于互联的导线中的电子在电流作用下往同一方向流动,最终造成互连线的截面面积变得不均匀,造成截面面积较小处形成开路或熔断,造成电路的失效。

寄生参数提取是指在完成芯片的布局布线后,提取器件和互连线的寄生电阻电容,用于芯片电学性能分析,包括时序、功耗、信号完整性分析等。寄生参数提取的准确性取决于器件参数模型和互连线模型。目前行业里最常用的模型是 SPICE 模型。

网表一致性检查(LVS)、电学规则检查(ERC)和几何设计规则(DRC)检查等三部分工作合称为物理验证。版图完成后,对版图上图形结构进行元器件的提取以及互连关系的提取形成新的网表,对该网表拓扑结构与原始网表的拓扑结构进行等价性检查,保证版图实现的电路与网表电路的一致性。电学规则检查主要检查版图是否存在天线效应、断开、短路等问题。几何设计规则检查主要检查集成电路版图中各几何图形尺寸,以及代表组成元器件不同部分的图形的组合是否符合加工设计规范,重点检查是否满足最小尺寸要求,如金属氧化物半导体(MOS)晶体管栅的宽度是否大于等于最小尺寸要求,金属对通孔覆盖的尺寸是否满足要求等。现代的设计方法中这些设计规则由 EDA 厂商和代工厂共同开发成为支持 EDA 工具的 Runset 文件,由 EDA 工具自动检查。

物理设计技术与集成电路工艺技术节点有密切的关系。在微米级(5 μm、3 μm、2.5 μm、1.5 μm 等)加工技术节点,物理设计重点在通过版图的布局布线尽可能获得最小的芯片面积。在亚微米、深亚微米(0.65 μm、0.35 μm、0.25 μm、0.18 μm)加工技术节点,物理设计重点在满足电路的时序,提高电路稳定性。信号完整性分析,基于时序驱动的布局布线技术是物理设计的关键。在超深亚微米工艺节点(0.13 μm、90 nm、65 nm),单芯片集成度大大增加,器件之间互连更加复杂,互连线寄生效应成为影响电路性能的主要因素,基于解决布线拥塞的布局布线技术、信号完整性分析、电迁移、电源网络噪声分析等成为物理设计的关键技术。到了纳米加工技术节点(45 nm、32 nm、28 nm、16 nm),考虑工艺涨落因素的可制造性设计(DFM 技术)、电路可靠性分析和优化技术成为纳米技术节点物理设计的关键技术。

参考文献

Scheffer L,等. 集成电路系统设计、验证与测试. 陈力颖,等译. 北京:科学出版社,2008（陈岚）

wulianwang

物联网(Internet of things)　一个基于互联网,让所有能够被独立寻址的普通物理对象实现互联互通,从而提供智能服务的网络。典型的智能服务包括识别、定位、跟踪、监控、管理等。1999 年,MIT 最初提出物联网的概念是把所有物品通过射频识别技术与互联网连接起来,实现智能化识别和管理。2005 年,ITU 物联网报告给出了物联网新的内涵:将各种信息传感设备,如射频识别装置、各种传感器节点等以及各种无线通信设备与互联网结合起来形成的一个庞大、智能网络,这样,所有的物品都能够远程感知,并与现有网络连接在一起,形成一个更加智能的物联网生态系统。

物联网具有以下三个重要特征:

(1) 普通对象设备化　即通过置入芯片、RFID、条码等手段使像茶杯、桌子、食品、轮胎、车辆等普通物理对象变成可寻址的设备;

(2) 自治终端互连化　即将上述设备化的物理对象作为网络自治终端进行连网;

(3) 普适服务智能化　在这个广泛互连的网络上,通过每一个普通对象参与服务流程,使普适服务智能化,如体载网络、车载网络中的传感节点能对司机身体和道路状况实时监测,从而指导驾驶行为。

与传统的信息网络相比,物联网具有新的目标,体现在三个方面,即更广泛的互连互通、更透彻的信息感知、更综合的智能服务。

按照网络分层的原理，可将物联网抽象成由环境感知层、数据传输层、服务支撑层、应用服务层构成的四层体系架构，其中环境感知层实现对监测环境中物理对象的感知和数据获取，数据传输层提供透明的数据传输能力，服务支撑层主要提供对网络获取数据的智能处理和服务支撑平台，应用服务层将信息转化为内容通过智能的终端向用户提供服务。

物联网的核心技术包括感知技术、组网技术、信息处理技术、服务提供技术、安全技术等。

参考文献

1. International Telecommunication Union. ITU Internet Reports 2005：The Internet of Things. November 2005

2. Huadong Ma. Internet of things：objectives and scientific challenges. Journal of Computer Science and Technology，2011，26(6)：1-7

（马华东）

X

xitong chengxu sheji yuyan

系统程序设计语言（systems programming language）　可以用于书写计算机系统程序的语言。

系统软件不同于应用软件，主要用于支持计算机本身的操作和使用，因此，它通常和运行该程序的计算机硬件结构有密切的关系，并且往往是常驻的，经常在起作用。因此，对系统程序设计语言有不同于一般高级语言的要求。一般高级语言力求实现与机器无关，以便隐蔽硬件结构，具有良好的易移植性。系统程序设计语言在保持高级语言优点的同时，特别强调程序的结构与功效，必须提供使用简便又兼顾实现功效的系统功能描述手段，有时就不得不包含少量与实现有关的成分。一种良好的系统程序设计语言应当在不牺牲易移植性的条件下实现上述要求。

在20世纪60年代以前，计算机系统程序都是用**汇编语言**或**机器语言**书写的。用**低级语言**书写系统程序生产率低下，质量控制和维护都很困难。因此，从50年代末开始研制能够书写编译程序，特别是能够书写自身的编译程序的高级语言。1958年出现的NELIAC语言具有开创性的意义。最初的系统程序设计语言产生的代码质量不高，曾引起争议。随着语言研制的进展，同时也由于软件规模的日趋大型化使得低级语言已难以满足程序人员的需要，自70年代初，高级语言取代了汇编语言，成为系统程序的主要描述工具。这期间出现的PL 360，BLISS，PASCAL，XCY，Modula-2，Edison以及Ada等语言均可以有效地用于书写系统程序。

目前，系统程序设计语言用于书写各种类型的系统程序，书写UNIX系统的**C语言**是使用最广泛的计算机**程序设计语言**之一。

参考文献

徐家福．系统程序设计语言．北京：科学出版社，1983

（陈道蓄）

xitong jianrongxing

系统兼容性（system compatibility）　为一种计算机系统开发的软件或硬件可适用于另一种或其他多种计算机系统的能力。

系统兼容性是系列计算机的基本特性，是在计算机硬件技术迅猛发展，市场竞争剧烈，生产厂商不断推出新产品型号的情况下，避免用户在老产品型号上开发的软件遭受废弃的一种重要设计思想与技术措施。它保护了用户的已有资源，节约了厂商和用户的开发投资，加快了计算机的研制过程，促进了计算机产业和应用的发展。

随着**计算机硬件**技术的发展，计算机的处理能力不断提高，应用领域急剧扩大，对软件的需求量不断增加，但软件的生产技术发展缓慢，开发效率低，周期长，费用高。如何使一种软件产品能尽量在多种计算机上使用是人们关心的问题，广大用户在要求更新计算机时，希望原有软件能在新的计算机上继续使用。计算机的生产厂家为取得更大的经济效益，更希望以巨资开发研制的软、硬件产品能用于多种计算机上。20世纪60年代由IBM公司开发的IBM 360系列计算机，可视为实现了系统兼容性的早期代表。在此之后，具有兼容性的系列机大量出现。

兼容性表现在软件和硬件的许多方面，兼容的范围和级别也各不相同，其实现方法分述如下：

（1）**机器语言**程序兼容　机器语言就是用硬件实现的机器指令。实现用机器语言编写的程序兼容对计算机体系结构有非常苛刻的要求，需要实现兼容的两台计算机的体系结构和操作系统的用户程序接口等应完全相同；即使略有不同，也可用软件模拟或硬件仿真实现兼容。但这些方法将使用户程序的运算速度明显降低。

（2）**汇编语言**程序兼容　汇编语言是一种低级的符号语言，是机器语言的符号化，与机器语言密切相关。因此，实现兼容所面临的问题与机器语言基本相同。但用汇编语言编写的程序要经过汇编程序汇编成机器语言程序才能在计算机上运行，这就给实现兼容提供了一些方便。汇编语言程序兼容的首

要条件是要在实现兼容的计算机上配有兼容的汇编语言文本及其汇编程序。如果语言文本不同，可以用转换程序解决，如用户程序与操作系统的接口差异可通过汇编程序转换解决，缺少某种指令可由汇编程序用广义指令来实现其功能等。但如果实现兼容的计算机体系结构差别较大，则汇编语言程序的兼容将难以实现。

（3）**高级语言**程序兼容　高级程序设计语言文本与机器的体系结构无关，所以用高级程序设计语言编写的程序容易实现兼容。这是用户程序实现兼容的主要方法。采用标准化的语言文本，则更容易实现兼容。但实际上，由于在不同计算机上对语言文本有不同的限制，可能影响兼容性的实现。

（4）系统软件兼容　系统软件是控制计算机运行和为用户程序服务的一类软件，主要包括**操作系统**、**编译程序**、**数据库管理系统**等。编译程序将用高级语言编写的源程序编译成机器语言程序，它与计算机的体系结构密切相关，因此，不同体系结构的计算机之间难以实现编译程序的兼容。操作系统是与计算机体系结构密切相关的系统软件，但从功能上看，操作系统可分为与计算机体系结构相关的部分和与用户程序相关的部分。前者构成操作系统的内核，不同计算机的操作系统有不同的内核，互不兼容；后者为操作系统的外层，与计算机体系结构无关，可以实现兼容。数据库管理系统也与计算机体系结构无关，可在操作系统的支持下，实现兼容。

（5）软件系统兼容　在软件的发展过程中，新的软件系统不断出现，因此也产生了各种软件系统之间的兼容问题。为了使在某种软件系统环境下开发的软件能在新的软件系统环境下正确运行，就需要新开发的软件系统与以前的软件系统兼容，如要求与某操作系统兼容，与某数据库管理系统兼容等。FoxPro 数据库管理系统就是采用向下兼容的技术与 dBASEⅢ，FoxBASE 等数据库管理系统兼容。各种计算机上配置的 UNIX 操作系统在外层上也有不同程度的兼容性。

（6）设备或部件兼容　设备或部件兼容是指一种设备或部件可不加改动地用于多种机器。这要求设备或部件符合某种标准化设计，包括设备或部件的功能、接口、约定、规范、规程等。

（7）系列机　系列机是计算机体系结构相同、具有标准的外围设备接口、软件向上兼容、性能和价格不同的大、中、小各种型号配套的一族机器。系列机既实现了软件兼容，也实现了硬件兼容，充分体现了兼容性的实现原则和优越性。

（8）兼容机　一些计算机厂家为了利用别人的软件成果，研制了兼容机。这些计算机体系结构可能不同，厂家也各不相同，但软件兼容，有的还实现了插件兼容。这种兼容机是选择一种市场前景较好的计算机作为兼容对象，按照这种计算机体系结构设计出可利用其软件资源的计算机。这种兼容机不但体系结构与兼容对象相同，甚至部件也是一样的。

随着计算机软、硬件技术的发展，设计人员更加注意产品的兼容性。实现兼容性的技术和方法会越来越受到人们的重视。但兼容性要求也往往给新产品的开发提出各种制约，降低了计算机体系结构变革的自由度。（时方绛）

xitong kongzhiqi xinpian

系统控制器芯片（system controller chip）　**微型计算机**系统中，与**中央处理器**（CPU）芯片直接相连的用于处理高速系统总线信号的桥接芯片。在表示微机主板原理的框图中，CPU 芯片一般被置于图的最上部，和它直接相连的系统控制器芯片位于图中其他部件的上方。沿袭地图绘制中上北下南的习惯，应用中常将系统控制芯片称为北桥（northbridge），将位于北桥下方的**外围控制芯片**称为南桥。

北桥是专门负责处理 CPU、主存、显卡和南桥之间通信的芯片。它既是 CPU 访问内存、显卡和南桥的通道，同时又是显卡、南桥访问内存的通道。由于北桥的主要功能围绕着内存展开，北桥又经常被称作访存控制中心（memory controller hub）。某些北桥芯片内部还集成了显示控制的功能，这类北桥又可以被看作是图形和访存控制中心（graphics and memory controller hub）。由于目前集成显示功能的北桥的图形计算能力与独立的显卡尚有差距，此类北桥通常保留了连接外部显卡的高速信号接口（如 PCIE 或 AGP 等）。由于不同的 CPU 芯片和内存芯片使用的接口信号具有差异性，一款北桥芯片通常仅与 1 个处理器厂家的有限特定系列的 CPU 芯片配套并仅支持 1 种规格的内存芯片。

图 1 展示了一幅典型的北桥结构框图。北桥通过高速总线分别与 CPU、南桥、内存、图形处理卡相连。CPU 通过北桥访问内存、图形处理卡和连接在南桥上的外围 I/O 资源。图形处理设备通过北桥访问内存。南桥通过北桥访问内存。在实际的北桥芯

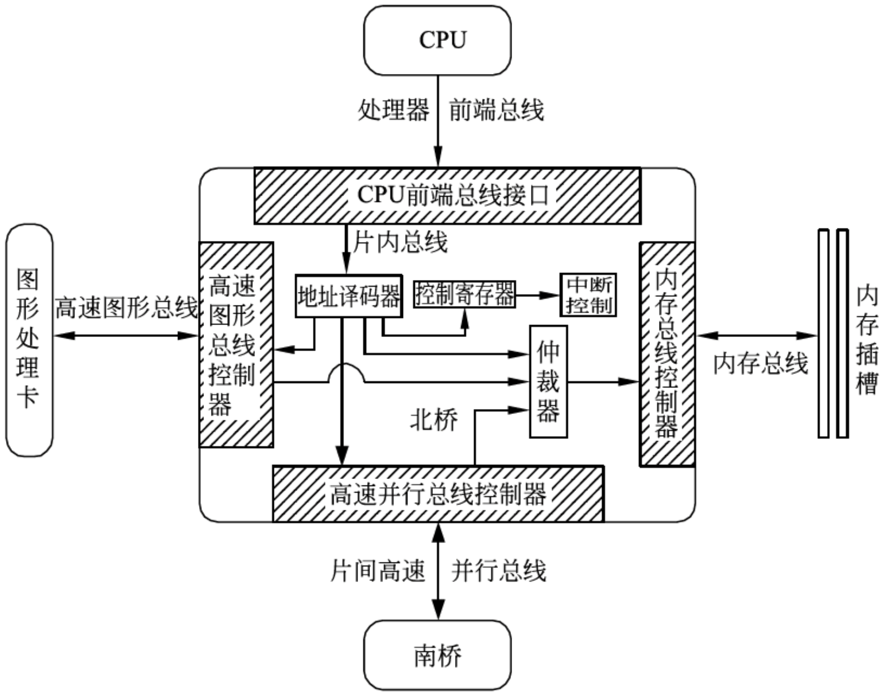


图 1　典型的北桥结构框图

片中,北桥与 CPU、内存、南桥互连的总线通常为高速并行总线;而北桥和图形处理设备相连的总线通常是以 AGP 为代表的高速并行总线或者是 PCIE 为代表的高速串行总线。北桥同时还为 CPU 提供整个系统的中断控制。

随着半导体工艺的进步,单位面积可集成的晶体管数目不断增加,北桥的功能正在越来越多地被集成进 CPU 之中以降低系统的总体成本并提高性能。在此趋势下,北桥将在**个人计算机**(PC)系统中逐步消失。取而代之的是由 CPU + 南桥或者直接以 CPU 为核心的单个芯片构成的 PC 系统。

参考文献

1. 陈国先. 计算机维护与维修. 3 版. 北京:机械工业出版社,2001

2. 杨全胜,等. 现代微机原理与接口技术. 2 版. 北京:电子工业出版社,2010

（蔡飞）

xitong xingneng zhibiao

系统性能指标(system performance criteria)　反映计算机系统性能和特征的一组参数。

不同的用户对系统性能指标的关心不同,例如科学计算的用户最关心 MIPS、MFLOPS、加速比;军事用户最关心可靠性和环境适应性;过程控制人员最关心实时性和可靠性;维护人员最关心可用性和可维护性。有关可靠性、可用性和可维护性参见**计算机系统可靠性**、**计算机系统可用性**和**计算机系统可维护性**。最常用的性能指标有 MIPS、MFLOPS、加速比、吞吐率、响应时间、利用率、性能价格比等。

通常,性能指标可分为绝对的和相对的两类。所谓绝对的是指不需要一定参照量的参数,如 MIPS 和 MFLOPS。所谓相对的是指相对于一定参照量的参数,如加速比是多结点并行处理速度相对于单结点处理速度的倍数。VAXMIPS 是相对于 VAX 11/780 系统的 MIPS 参数。

性能指标又可分为基本的和导出的。所谓基本的是指不能由其他参数导出的参数,这种参数是直接获得的,如处理时间 T。所谓导出参数是指通过基本参数而间接得到的,如 MIPS,MFLOPS 分别是由指令数和浮点操作数与处理时间导出的;加速比是由多结点处理时间和单结点处理时间导出的。

性能指标还可分为工作量类、响应性类和利用率 3 种。吞吐率、工作速度、指令执行速率和数据处理速率属工作量类。响应时间、周转时间和反应时间属响应性类。硬件(CPU,内存,I/O 通道,I/O 设备等)的利用率,操作系统的利用率,公用软件(如编译程序等)的利用率,数据库的利用率属利用率类。

MIPS 表示百万[条]指令每秒,用来描述计算机的定点运算速度。常用的有峰值 MIPS、基准程序

MIPS 和以特定系统为基准而得到的 MIPS。对于 1 个处理机芯片或 1 台串行计算机，其峰值 MIPS 值通常是以其指令集中最基本指令的执行速度而计算得的。如设某一系统的指令集中最基本指令的执行需 a 个机器周期，每一机器周期为 t 微秒，则其峰值 MIPS 值为 $1/(at)$。一台处理机的平均 MIPS 值是指按照一定的加权值平均其指令集中各类指令的执行速度而计算得的 MIPS 值。基准程序 MIPS 值是指用基准程序测得的 MIPS 值。对于某一计算机，所用基准程序的不同，测得的 MIPS 值也不同，这主要是由于不同的基准程序中各种指令的混合比例不同以及指令集中各种指令的执行速度不同而引起的。MIPS 指标用于评价同一厂商生产的同一系列的计算机的定点运算速度比较准确，因为这些计算机有相似的系统结构、操作系统、语言和编译器，尤其是相似的指令集。如果两台计算机的结构和指令集不同，那么用 MIPS 值来比较它们的运算速度是不准确的，有时可能会导致错误的结论。

MFLOPS 的定义是百万[次]浮点运算每秒，用来描述计算机的浮点运算速度，衡量计算机的科学计算性能。MFLOPS 的定义可分为两种：峰值 MFLOPS 和以基准程序测试得到的 MFLOPS。对于单个处理机芯片或一台串行计算机，其峰值 MFLOPS 通常是以其最快的浮点操作速度或以各浮点操作的平均速度计算得到的。设其最快的浮点操作的执行速度或各浮点操作的平均速度为每个机器周期 a 次，每个机器周期为 b 微秒，则其峰值 MFLOPS 定义为 a/b MFLOPS。串行计算机的 MFLOPS 值也可用基准程序测得。设基准程序 A 在其上执行的时间为 t 微秒，程序 A 中含 F 个浮点操作，则其 MFLOPS 为 F/t MFLOPS。F 的计算有不同的方法。有的是直接统计机器所执行的浮点操作次数，有的是根据问题的复杂性分析，从数学模型中计算出来。前者获得的是机器实际执行的浮点操作数，其中，可能包括一些辅助性的非问题本身的浮点操作，而后者只包括与问题有关的浮点操作。MFLOPS 可用于比较和评价在同一系统上求解同一问题的不同算法的性能。MFLOPS 还可用于在同一源程序、同一编译器以及相同的优化措施、同样的运行环境下，利用相同的方法对不同系统测试浮点运算速度。由于实际程序中各种操作所占比例不同；不同浮点操作的运算速度不同；对于传统的超级计算机，MFLOPS 值没有考虑向量化程度对运算速度的影响；对于并行处理系统，MFLOPS 值没有考虑诸如通信、竞争共享资源等对运算速度的影响；MFLOPS 值没有考虑运算部件与存储器、I/O 系统、互联网络等速度之间的相互协调等因素。所以 MFLOPS 值只能近似地说明在特定条件下系统的浮点运算速度。

加速比是度量多结点并行处理比单结点处理的加速倍数，用来描述并行处理的效果。绝对加速比以当前解决问题的最好串行算法作为比较基准，它着眼于并行处理相对于串行处理的优化效果上，用于评价并行算法。相对加速比定义为同一并行算法在单结点上运行时间与多个相同结点构成的并行处理系统上运行时间之比。它着眼于并行算法和计算机本身的可扩展性，用于评价计算机系统的性能。

吞吐率指计算机在单位时间内能处理的信息量。响应时间指从给定计算机输入到出现对应的输出之间的时间间隔。

响应时间取决于用户输入的信息、系统特性以及在用户输入信息时系统正在处理的其他负载。虽然响应时间是一个随机变量，但在相当长的时间周期内可以用统计规律描述它。

利用率指在给定的时间内，计算机某一部分的实际使用时间所占比例。

在事务处理领域中，常用事务处理每秒(TPS)来表示计算机的处理速度，但“事务”没有统一的定义。部分计算机厂商以[次]运算每秒(OPS)表示计算机的处理速度。

性能价格比也是常被用作评价计算机的参数之一。它以每元人民币或每美元所能买到多少 MIPS 或 MFLOPS，或以每 MIPS 或 MFLOPS 值多少人民币或美元来表示。

上述系统性能指标一定程度上反映了计算机系统的性能和特征，但有时不够精确，甚至有时会得出错误的结论。必须注意，上述系统性能指标均与工作负载以及系统特性有关，评价者必须清楚是在什么系统和什么样的负载情况下测得的性能指标，否则，是没有意义的。　　　　（郑纬民）

xitong zongxian

系统总线(system bus)　计算机系统中，在多于 2 个模块(部件或子系统)之间传送信息的公共通路。

计算机采用系统总线不仅可以大大简化硬件的设计过程，简化系统结构，使系统易于扩充，而且能

简化系统的软件设计过程,减轻软件的设计和调试工作负担,从而缩短了软、硬件的研制周期,降低了系统的成本。

在一个计算机系统中,按照规模、用途、在系统中的层次及其应用场合的不同,总线可分为3类:

(1) 片总线　又称"芯片总线",是处理器芯片引出的信号线,它是用处理器芯片构成一个部件(如CPU插件)或是一个很小的系统时,构成部件(或小系统)的各元、器件之间信息传输的通路。是"元件级总线"。

(2) 内总线(I-BUS)　又称"系统总线"或"板级总线",是计算机系统内部扩展总线,它是用于构成计算机系统各插件(板卡)之间信息传输的通路,是模块级总线。

(3) 外总线(E-BUS)　又称"通信总线",它是计算机系统之间,或是计算机系统与其他系统(仪器、仪表、控制装置)之间信息传输的通路,是系统级总线,往往借用电子工业其他领域已有的总线标准。

系统总线是计算机系统内部各插件板之间传输信息的公共通路。为了在各模块之间实现信息共享和交换,总线由传输信息的信号线及一套管理信息传输的通用规则(协议)所构成。

系统总线的信号线大致可分为5类:①数据传输信号线,包括地址线、数据线以及读-写控制线等;②中断信号线,包括中断请求线、中断认可线等;③总线仲裁信号线,包括总线请求线、总线许可线和总线忙线等;④其他信号线,包括系统时钟、复位、电源和地线等;⑤备用线,用于扩充功能或特殊用途。近30年来比较有影响的系统总线有IBM PC/XT总线、ISA总线、EISA总线、MULTIBUS、VME、VME64总线、MCA总线、FutureBus和FutureBus+总线、STD和STE总线、VL-Bus、PCI、PCI-X总线、PCI-E总线等。

系统总线最基本的任务是保证信息在总线上可靠地传输。为了使信息源和信息接收部件能同步,在总线上传输信息时必须遵守一定的定时规则,这种定时规则称为总线定时协议。总线定时协议通常分为3种:①同步总线定时协议,所有模块都连接到公共时钟上,总线操作都在公共时钟控制下的固定时间内进行;②异步总线定时协议,总线操作由信息源或信息接收部件的特定跳变所确定;③半同步总线定时协议,总线操作之间的时间间隔按公共时钟周期的整数倍变化。

系统总线的一个重要的性能指标是它的带宽。**总线带宽**是总线本身所能达到的最高传输速率,所以又称为总线数据传输速率,其单位是MB/s或Mb/s。与总线带宽密切相关的两个因素是总线的位宽和总线的工作频率,例如,EISA总线的数据宽度为32,工作频率为8.33 MHz,其最大带宽为33.32 MB/s。总线的实际带宽还受3种因素的制约:①挂在总线上的模块数量;②总线布线的长度;③总线驱动器和接收器的性能。

随着微电子技术和计算机系统设计技术的迅速发展,系统总线也在不断地发展和完善,从性能和技术上看,目前总线向以下几个方面发展,一个是支持工业控制应用的总线,要求具有强有力的工业输入输出支持,组合灵活,价格低廉,一般只要求支持8位和16位微处理器,如STD总线、STE总线等。另一个是提高系统处理能力的总线,如要求能支持32位和64位处理器以及支持高速网络传输和多媒体传输等,属于这一类的总线有PCI总线、PCI-E总线和HT总线以及工作站中的InfiniBand总线等。还有一类是适应便携式计算机的小型板卡的总线,如PCMCIA等。

参考文献

1. 金兰,金波. 计算机组织:原理、分析与设计. 北京:清华大学出版社,2006

2. 孙德文. 微型计算机技术. 3版. 北京:高等教育出版社,2010

(孙德文)

xibao zidongji

细胞自动机(cellular automaton)　并行计算机的一种理论模型。细胞自动机可视为由若干小单元构成的动态阵列,其中每一单元具有有限个状态,在离散步序中,每一小单元按一致的法则,由其原状态及其邻域单元的状态决定新的状态。在任一时刻,诸小单元状态的总体构成细胞自动机的格局。从初始格局到最后格局的演化过程为计算处理过程,最后格局被视为计算结果。实际上,细胞自动机的每一小单元均为一有限态自动机,细胞自动机即为有限态自动机的动态阵列,细胞自动机格局的演化过程即为并行计算过程。细胞自动机的主要功能在于:可由局部特性及简单的一致性法则模拟、处理总体上具有高复杂性的离散过程和现象。所以,细胞自动机除了在计算机科学中的重要意义外,它还是描述现实世界中大型复杂离散系统的数学工具。

细胞自动机的“小单元”在细胞自动机结构中称为细胞。一个细胞自动机定义为有规律地分布于 n 维空间中的一个细胞空间或细胞集合。$n=1$ 是最简单情形，细胞分布在一条直线上，每一细胞为一方格，每一细胞有两个邻域细胞（简称邻域），每一细胞仅有两个不同状态，可由 0 和 1 值表示。在每一时刻，全部细胞状态值构成的序列为细胞自动机在该时刻的格局。在 $n=2$ 的情形，细胞在二维平面上分布，它们可为连续的方格、等边三角形、蜂房形式或其他形式。一个简例是，细胞空间为 z^2，其中 z 为整数。每一细胞 $x=(x_1,x_2)$ 直接连接它的 8 个邻域，每一细胞的状态值 $q\in\{0,1\}$。局部性法则为局部传递函数 δ：

$$\delta(q_0,q_1,q_2,\cdots,q_8)=\begin{cases}1,\text{若 }q_0=0\text{ 且}\sum_{i=1}^{8}q_i=3\\ \quad\text{或 }q_0=1\text{ 且 }2\leqslant\sum_{i=1}^{8}q_i\leqslant 3,\\ 0,\text{其他情形}\end{cases}$$

其中 q_0 为所考虑的细胞的状态。可知，细胞在离散步序中的新值由该细胞的当前值及其 8 个邻域细胞的当前值按法则 δ 确定。在每一时间步序，δ 以并行方式同步地作用于每一细胞状态，从而实现细胞自动机总体格局的并行计算。一般地，细胞自动机的基本模型具有五个主要特征：

（1）它们由细胞的离散格局构成；

（2）它们在离散时间步序中演化；

（3）每一细胞的状态均在同一有限集中取值；

（4）每一细胞的状态依同一确定的法则演化；

（5）细胞状态的取值法则仅依赖于其自身及其周围局部领域细胞的状态值。

由于不同的目的和需要，出现了许多细胞自动机的变异模型。这些变异由以下因素产生：细胞相互联接形式的改变；细胞取值随机性的引入；细胞间信息交换（通信）机制；外部信息的输入及信息输出，等等。这使得在文献中也使用了其他一些术语。

细胞自动机提供了分布并行计算系统的数学模型，它在新一代计算机结构设计中有重要意义。细胞自动机在模式识别、图像处理及人工智能中有重要应用，在计算机视觉研究中是合宜的模型。细胞自动机的理论研究，涉及计算理论中众多深层次问题。

实际上，细胞自动机的理论和应用不仅限于计算机科学本身。由于细胞自动机可描述具有很大自由度的离散动态系统，它可视为偏微分方程离散化的理想形式。所以，它在物理学中有广泛应用，特别是对非线性问题的处理。细胞自动机的另一些重要应用是在生物学中，涉及到生命进化过程、神经网络等方面。作为其逻辑结构和功能可被细胞自动机阐明的一个生物系统，D. Farmer 等人给出了研究 DNA 序列的一个新方法，指出，DNA 序列的演化，遗传信息的存储和更新，细胞自动机作为模型是适宜的。

在现实世界中，自然现象的许多个别或局部规律人们是易于了解的，但构成现象的整体特性往往非常复杂，难以把握。细胞自动机提供了研究这类现象的一种手段。一个简单例子是细胞自动机可模拟雪花生成的过程和机制。对细胞自动机格局演化过程中极限集的研究，可发现分数维问题和混沌现象。细胞自动机是研究混沌理论的数学工具之一。

细胞自动机理论和应用的研究不过短短 40 余年，处于开始阶段。伴随计算机科学的发展，它的作用将会日益显现。

参考文献

1. Farmer D, Toffoli T, Wolfram S. Cellular automata. Amsterdam: North-Holland, 1984

2. Demongeot J, Golès E, Tchuente M. Dynamical systems and cellular automata. London: Academic Press, 1985

3. Toffoli T, Margolus N. Cellular automata machines. Cambridge, MA: MIT Press, 1987（周广福）

xifen qumian

细分曲面（subdivision surface） 基于一组拓扑规则和几何规则对初始多边形网格 M^0 递归地细化与平滑所得到的网格序列 $\{M^k\}$ 的极限曲面（图 1）：

$$M^\infty=\lim_{k\to\infty}M^k$$

网格细化所遵循的规则称为细分模式。其中，拓扑规则确定新网格顶点的插入方法及连接关系，几何规则定义新网格中顶点位置的计算方法。细分过程中，控制网格顶点位置不变的模式称为插值细分，否则称为逼近细分。

细分方法可追溯到 1947 年 G. de Rham 生成光滑曲线的角切削思想。1974 年 G. Chaikin 给出了第一个细分曲线生成模式。E. Catmull 和 J. Clark 于

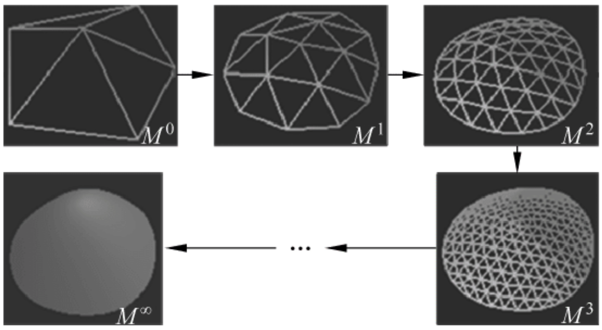


图 1　细分曲面生成过程示意图

1978 年从双三次 B 样条的递推性质导出任意拓扑网格上的细分曲面模式。同一年提出的 Doo-Sabin 细分曲面则是双二次 B 样条的推广。1987 年提出的三角网格上的 Loop 细分，则是四次三角箱样条（box spline）在任意拓扑网格上的推广。

Doo 和 Sabin 通过对细分矩阵进行特征分析，对细分模式的收敛性和细分曲面的连续性进行了探索，建立了细分曲面性质研究的核心技术。严格的连续性理论则由 U. Reif 于 1995 年建立，D. Zorin 等人于 1998 年完善。M. Halstead 和 H. Hoppe 等分别发现了 Catmull-Clark 细分和 Loop 细分的控制顶点极限位置的解析表达。J. Stam 于 1998 年进一步提出 Catmull-Clark 与 Loop 细分曲面的精确计算方法。

由于细分曲面思想简洁，能表示复杂的几何形状，具有很好的多分辨率性质，在图形学，特别是在场景复杂但精度要求不高的计算机动画与三维游戏领域得到了广泛应用。

虽然已提出了大量细分曲面模式，细分曲面的性质及不同模式之间的关系值得进一步研究。为了满足各类应用的需求，在曲面属性**图形处理器**快速计算、高效的形状控制、多分辨率编辑、高精度的细分曲面求交和裁剪及基于细分曲面的逆向工程等领域仍有许多问题亟须解决。

此外，处处二阶连续的细分曲面构造尚无令人满意的结果。动态细分曲面构造及其连续性分析框架有待建立。对体网格细分的相关理论与细分模式构造所知不多。此外，顶点位置附有其他属性的更一般的细分仍有发展的空间。

参考文献

1. Warren J, Weimer H. Subdivision methods for geometric design—a constructive approach. San Francisco, CA: Morgan Kaufmann Publishers, 2002

2. Peters J, Reif U. Subdivision surfaces. Springer, 2008

3. Zorin D, Schröder P. Subdivision for modeling and animation. In: Proceedings of ACM SIGGRAPH 2000 Course Notes #23, 2000

（李桂清）

xiatui zidongji

下推自动机（pushdown automaton）　带有栈的**有限自动机**。该栈表是一个存储量没有限制的辅助存储器，是一个“先进后出”的栈。存入一个符号（压栈）就把从前放入栈里的符号顺次往下推一次，取出一个符号（弹出）必定是取栈顶符号，栈内其他符号顺次向栈顶移动一次。取放重叠于一弹簧上的盘子就是弹出压栈的一个形象表示。下推自动机主要讨论由**上下文无关文法**产生的上下文无关语言的关系、下推自动机的类型和子类、程序语言语法的确定型下推自动机、下推转换器和不可判定问题等。

下推自动机各种变形及其子类　下推自动机（PDA）的数学定义为七元组 $M=(Q,\Sigma,\Gamma,\delta,q_0,Z_0,F)$，其中有限集 Q 为状态集，有限集 Σ 为输入字母表集，Γ 为栈符号集，$q_0\in Q$ 称为初态，$Z_0\in\Gamma$ 是一个称为栈开始符号的特殊符号，$F\subseteq Q$ 称为终态集，δ: $Q\times(\Sigma\cup\{\varepsilon\})\times\Gamma\rightarrow Q\times\Gamma^*$ 是多值函数。例如

$$\delta(q,\varepsilon,Z)=\{(p_1,\gamma_1),(p_2,\gamma_2),\cdots,(p_m,\gamma_m)\}$$

表示 PDA 在状态 q 时与扫描的输入字母无关，栈顶符号为 Z，可以转移到状态 $p_1,p_2,\cdots$，或 p_m，并以 $\gamma_1,\gamma_2,\cdots$，或 γ_m 相应于 $p_1,p_2,\cdots$，或 p_m 代替 Z，且输入头不动。定义表明，下推自动机是一个不确定的机器。$L(M)=\{x\in\Sigma^*\mid\delta(q_0,x,Z_0)=(p,\varepsilon,\gamma),p\in F,\gamma\in\Gamma^*\}$ 称为下推自动机按终态接收的语言。研究处理这类语言的算法时，必须准备好从前面错误选择转移动作里恢复正确选择的办法，但这往往是很麻烦的，为了避免这种情况，引入下推自动机按空栈接收的语言，它的定义为 $N(M)=\{x\in\Sigma^*\mid\delta(q_0,x,Z_0)=(p,\varepsilon,\varepsilon),p\in Q\}$。已经证明下推自动机 M_1 按终态接收的语言必为某一下推自动机 M_2 按空栈接收的语言，且任一 M_2 按空栈接收的语言必为某一 M_1 按终态接收的语言，即 $L(M_1)=N(M_2)$，PDA M_1 与 M_2 等价。PDA 与**形式语言理论**中**乔姆斯基分层**的**上下文无关语言**等价，即任一上下文无关语言必有一个 PDA 接收它，且 PDA 接收的语言必为上下文无关语言。任一上下文无关语言状态数最多可以为 2、无 ε-转移的按终态的 PDA 所接收，也可以为 $\delta(q,a,Z)$ 的栈符号不超过 2 的 PDA 所接收。

满足条件：对每一 $q \in Q$，每一 $Z \in \Gamma$，(1) 当 $\delta(q,\varepsilon,Z) \neq \varnothing$ 时，对所有 $a \in \Sigma$，$\delta(q,a,Z) = \varnothing$。(2) 对所有 $a \in \Sigma \cup \{\varepsilon\}$，$\delta(q,a,Z)$ 的元数不多于 1 的 PDA 称为确定下推自动机(DPDA)。条件(1)说明与输入无关的转移和与输入有关的转移不会产生混淆，避免了扫描一个输入字母和出现使用空字的自发转移的不确定性。条件(2)说明 δ 为单值函数或单值偏函数，从而表明了确定性。若 DPDA 的栈操作只有擦去栈顶符号或压入一个栈符号，则称该 DPDA 为标准形。任一DPDA存在一个等价的 DPDA，它扫描任一输入字的全体字母。带 ε-转移的 DPDA 和不带 ε-转移的 DPDA 的功能不同。又，DPDA 类是 PDA 的真子类，如语言 $\{a_1a_2\cdots a_na_n\cdots a_2a_1 \mid a_i \in \Sigma\}$ 可以为一个 PDA 所接收，但不能被任何 DPDA 所接收。允许输入头在输入带的两个方向移动的 PDA 称为双向下推自动机(2PDA)，它接收在终态时移动离开右端的输入串。语言 $L = \{0^n1^n2^n \mid n \geqslant 1\}$ 可被一个 2PDA 所接收，但不能被任何 PDA 所接收，故 PDA 与 2PDA 不等价。L 是某一 DPDA 所接收的语言，如果 $x \in L$，x 的任何真前缀都不在 L 中，称 L 具有前缀性质。已证明具有前缀性质的上下文无关语言 L 可由某一LR(0)文法（参见 **LR(k)文法**）所生成。为了给许多程序语言提供方便而自然的语法分析生成器，增加朝前看一个符号的 DPDA(对应于 LR(1)文法)就能识别更多的语言，因而 DPDA 对编译设计极为重要。

下推自动机的运算、预测机和不可判定问题 PDA 经替换、同态映射、逆同态映射后仍为 PDA。PDA 与有限自动机之交、PDA 对有限自动机的商，求 INIT、CYCLE、reversal 仍为 PDA。但两个 PDA 之交、PDA 之补不再是 PDA。DPDA$M = (Q_M,\Sigma,\Gamma,\delta_M,q_0,Z_0,F_M)$ 与有限自动机 $A = (Q_A,\Sigma,\delta_A,q_0', F_A)$ 产生的机器 $\Pi(M,A) = (Q_M,\Sigma,\Gamma \times Q_M \times Q_A,\delta,q_0,x_0,F_A)$ 称为预测机，其中 $\delta: Q_M \times \Sigma \times (\Gamma \times Q_M \times Q_A) \to Q \times \Gamma^*$，$x_0 = [Z_0,\mu_0]$，且 δ 和 μ_0 满足若干附加条件。用预测机可以得到 DPDA 若干运算性质，如 DPDA 在补、MIN、MAX 运算下仍为 DPDA，DPDA 与有限自动机之交、对有限自动机的商仍为 DPDA，但在同态映射下，或经并、连接、克林闭包运算后就不再是 DPDA 了。PDA 是不是一个 DPDA，两个 DPDA 的交是否为空，两个 DPDA 之交、并是不是 DPDA，一个 DPDA 是不是另一 DPDA 的子自动机等都是不可判定的。但有算法判断一个 PDA 所接收的语言是否为空、是否为有限，且有 CYK 算法判断一个字是否为一个给定的 PDA 所接收。

下推转换器 数学定义为 $M = (Q,\Sigma,\Gamma,\delta,\Delta,q_0,Z_0,F)$，其中有限集 Q 是状态集，有限集 Σ 是输入字母集，Γ 为栈符号集，有限集 Δ 为输出字母集，q_0 为初态，$F \subseteq Q$ 为终态集，$\delta: Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \to Q \times \Gamma^* \times \Delta^*$ 是转移函数。机器从 q_0 出发，输出带为空，栈顶符号为 Z_0，输入串 x 在输入带上，机器运行时，从左到右逐次扫描 x 的每个字母，δ 根据当前状态 p，当前输入带上的内容 a，当前栈顶符号 Z 规定出允许的结果(q,y,γ)，其中 q 为下一状态，y 为输出带上面下一段的输出字，栈顶的符号串 γ。下推转换器与下推自动机的运行相似，但另外生成一个输出串作为处理的一部分。M 用终态方式生成的翻译集合 $T(M) = \{(x,y) \mid x \in \Sigma^*, y \in \Delta^*, y$ 在输出带上，M 从 q_0, Z_0 开始，扫描完 x，M 在 F 的一个状态时停止运行$\}$。用空栈方式生成的翻译集为 $T_\varepsilon(M) = \{(x,y) \mid x \in \Sigma^*, y \in \Delta^*, y$ 在输出带上，栈空时停止运行，接收 $x\}$。已经证明，用空栈方式生成的翻译集当且仅当它是上下文无关句法制导翻译模式(SDTS)系统的翻译集。由 SDTS 构造的下推转换器直接用于模式识别，如描述亚中央染色体和远端染色体串的染色体文法。下推自动机模拟可接收输入串的最左派生(导出)，是分析上下文无关语言自顶到底分析器的标准模型，当出现可供选择的分支时，下推变换器总是选择正确动作，这个假定在数学上是虚构的，但是这种虚构使得不必逆向跟踪或并行推导，使问题简化。对下推变换器引入算子(将 Σ 的元素变为其他字母的错误)可以校正翻译中的错误，并已推广到概率自动机的研究中。

参考文献

1. Hopcroft J E, Ullman J D. Introduction to automata theory, languages, and computation. Reading, MA: Addison - Wesley, 1979

2. Gonzalez R C, Thomason M G. 句法模式识别. 濮群，徐凤家，徐光佑，译. 北京：清华大学出版社，1984

（张一立）

xianshiqi

显示器(display device) 由监视器、显示适配器(显示卡)和有关电路组成的用以显示字符、图形和图像的输出设备。向计算机输入的各种信息，计算机运算和处理的结果，都通过显示器转换成人眼可见的字符、图形和图像。

显示器的基本术语和主要技术指标

(1) **亮度**　指荧光屏发光的等级。亮度可分为4级,即暗、淡、亮和特亮。画面的亮度和显示点的发光强度、发光时间两者大体成正比。一般在室内较亮的环境下,显示器亮度应大于120 cd/m^2(坎每平方米)。

(2) **对比度**　指荧光屏画面上最大亮度与最小亮度之比。显示器对比度一般在300: 1 和10 000: 1之间。

(3) 屏幕尺寸　习惯上用屏幕对角线的长度表示。

(4) **像素**　屏幕上能够独立控制其颜色和亮度的最小区域。计算机对图像信息的处理是以像素为基本单位的。

(5) 点距　显像管屏幕上相邻两个像素之间的距离。点距数值越小图像越清晰。

(6) **分辨率**　衡量图像细节表现力的技术参数。通常用单位面积显示像素的数量来表示,即由水平方向的像素个数和垂直方向的像素个数的乘积来表示。如640×480,表示水平方向的像素个数是640个,即有640列;垂直方向的像素个数是480个,即有480行。常见的分辨率有1024×768,1600×1200, 2048×1536,2560×2048等。

(7) 能耗　显示器在不同工作状态下的能耗是不同的,通常在待机状态下的能耗较小。

(8) 可视角度　在纵横方向可以看到图像的最大角度。

(9) 反应时间　一个像素从黑到白,再回到黑状态所用的时间。数值越小越好。

显示器分类

显示器的种类繁多,可以按不同标准来分类。

(1) 按显示器件分:①发光器件　如阴极射线管(CRT)、等离子体(PDP)、发光二极管(LED)、电致发光管(ELD)等。这些器件在外加电信号后本身会发光。②光调制器件　如液晶显示(LCD)、电化学反应显示(ECD)等。这些器件本身不发光,但在电信号作用下,介质的光学特性起变化,使光线透过或反射,组成人眼可见的图像。

(2) 按所用显示适配器分:MGA显示器、CGA显示器、VGA显示器、SVGA显示器、多频显示器等。

(3) 按屏幕尺寸分:常用的有35 cm(14 in),43 cm(17 in),50 cm(21 in)等。

(4) 按显示颜色分:①单色显示器　只能显示两种颜色。可显示的颜色有白色、黑色、橘红色、琥珀色、绿色等。这类显示器体积小、重量轻、价廉,适用于流动性较强的场合。②彩色显示器　能够显示许多种颜色。

(5) 按输入信号形式分:①输入为数字信号　这类显示器的输入信号是分离式的晶体管逻辑[电路](TTL)脉冲信号,MGA及CGA显示器属于这一类。②输入为模拟信号　这类显示器输入信号是3个模拟信号,理论上它可显示无穷多种颜色,但实际的彩色种类还受显示适配器的控制。VGA及SVGA显示器属于这一类。

彩色阴极射线管(CRT)显示器

自然界中的各种颜色都可以由相互独立的三种单色光以适当的比例混合得到,这三种颜色被称为三基色。其中任一种基色不能由另外两种基色混合而得到。在显示器中用红、绿、蓝作为三基色。彩色显示器的工作是基于三基色原理。先把彩色图像分解成红、绿、蓝三种基色图像,然后分别转换成电信号,传送到CRT显示器后,再合成原来的彩色图像。

CRT显示器的基本结构是由CRT监视器和CRT显示适配器组成。

(1) CRT监视器　由阴极射线管、视频放大电路、光点定位电路(行扫描电路、场扫描电路、同步电路)和电源等部分组成。

阴极射线管又称**显像管**,是将电信号转变为可见图像的电真空器件。可分为黑白显像管和彩色显像管两大类。彩色显像管有产生红、绿、蓝三种基色的荧光屏和激励荧光屏的三个电子束。由三基色荧光粉所产生的分量不同的光来形成各种颜色。光点定位方法广泛应用的是光栅扫描。电子束在荧光屏上从左到右、从上到下一行行地有规律地扫描,形成一组光栅。根据需要显示的信息对电子束进行调制,形成图形或有明暗层次的图像。电子束过后,发光很快衰减而后消失。为了能看到稳定的图像,就必须不断重复扫描。通常电子束在屏幕上运动是从左到右、从上到下。沿水平方向的扫描称为行扫描,沿垂直方向的扫描称为场扫描。相应的扫描频率分别称为行扫描频率(行频)和场扫描频率(场频)。

CRT显示器扫描图像的方法有两种。一种是将一幅画面一行行地依次扫描,称作逐行扫描。计算机用的CRT显示器多用逐行扫描。另一种是将一幅画面分成两半,由奇数行组成奇数场,偶数行组成偶数场。先扫奇数场,再扫偶数场。两画面镶嵌在一起产生的视觉还是一个完整的画面,这种扫描方法称作隔行扫描。标准电视制式采用的是隔行

扫描。

场频是电子束从左上角到右下角的扫描速度。扫描速度越高,图像越稳定。当场频大于 24 Hz 时,人眼就有连续感;场频低于 47 Hz 会有明显的闪烁感;场频高于 75 Hz 才不会有闪烁感。CRT 显示器的场频在 60 ~ 180 Hz 之间。

行频是电子束从左到右的扫描速度。显示器的场频和行频有两类:①固定频率的显示器只有一种场频和行频。显示卡的频率必须与显示器一致才能使用。②多频自动跟踪显示器的场频和行频可在一定范围内变化,能支持多种 VGA 卡。这是常见的类型。对于逐行扫描,行频 = 1.1 × 场频 × 垂直分辨率。CRT 显示器的行频在 30 ~ 120 kHz 之间。

CRT 显示器具有响应速度快、分辨率高、使用寿命长、尺寸大、色域宽、颜色响应准确等优点,适合于出版、绘图等应用。它是应用较多、技术比较成熟的显示器。但由于它体积大、重量大、功耗大、有辐射、易受外来磁场干扰等缺点,在许多应用场合被液晶显示器(LCD)所取代。

(2) CRT 显示适配器　连接计算机的总线(参见**系统总线**)与 CRT 监视器以实现对监视器控制的接口部件。

显示适配器通常组装成电路板,直接插在**微型计算机**总线上使用,故又称显示卡。出于成本和体积考虑,也可以将显示适配器集成到主板上。

显示适配器一方面通过主机板上的扩展槽与主机系统总线连接;另一方面通过多芯电缆将视频信号、亮度信号、垂直和水平同步信号等送往显示器。它由寄存器、显示存储器、**只读存储器**、显示处理器和接口电路等部分组成。移位寄存器把输出数据变成串行数据,经视频电路转换成视频信号。显示存储器保存要显示的图像数据,即屏幕上的每一个像素的亮度和颜色信息。每个像素占用的位数越多,则能表示的色彩种类和亮度的层次也越多。例如每个像素只用一位,不是 1 就是 0,像素也只有两种颜色,不是黑就是白。用 8 位表示,有 $2^8 = 256$ 种颜色;用 16 位表示,有 $2^{16} = 65\,536$ 种(通常称为高彩色);而用 24 位表示,则有 $2^{24} = 1.67 \times 10^7$ 种颜色(称为真彩色)。通过改变显示存储器中的内容,就可改变屏幕上显示的图像。

显示适配器有不同的型号和规格,必须与相应规格的监视器配套使用。

液晶显示器(LCD)

一种本身不发光的显示器件。通过对环境光的反射或对外加光源控制的方式来显示图像。液晶介质发现于 1888 年,它在一定温度范围内既有晶体所特有的各向异性,又具有液体所特有的流动性。利用它在电场作用下能沿电场的方向排列成行,透光率随电压改变的特性,1968 年 RCA 公司制成了液晶显示器。1973 年,夏普公司把它用于电子表和计算器。

(1) 液晶显示器种类　从物理效应上分有动态散射型和扭曲向列型。动态散射型多用于仪器、仪表和计算器,扭曲向列型多用于计算机显示器。从结构上可分为笔段型和点阵型。笔段型多用于仪表,点阵型用于计算机显示器。

(2) 液晶显示器的基本结构　由两片刻有透明导电电极的平板玻璃基板夹着一个液晶介质层组成。

液晶本身不发光,它需要一个亮度高且均匀的背光源。彩色液晶显示器上使用最广泛的背光源是冷阴极荧光灯(CCFL)。在最底层和灯在一起的还有导光板和增亮膜,前者把线光源雾化成均匀的面光源,后者把光线聚拢使其垂直进入液晶层。在光源上面是玻璃基板,基板上面有偏振膜和驱动矩阵薄膜晶体管(TFT)电路。然后紧贴着的是液晶层,在液晶层上面,除了有玻璃基板、偏振膜和驱动矩阵薄膜晶体管电路之外,还有滤色片。

具有扭曲向列效应的液晶在未加电场时,其分子排列平行于电极表面,但在上表面与下表面之间旋转 90°。这种旋光特性在外加电场作用下会减弱或消失。如果液晶层上下表面的偏振膜的光轴是平行的,那么不加电场时由于液晶的扭曲效应,光线通不过偏振膜,因而像素对应的点是暗的。加电场后液晶不发生扭曲,光线通过偏振膜,像素对应的点是亮的。如果偏振膜的光轴是互相垂直的,不加电场时点是亮的;加电场后点是暗的。驱动矩阵电路中的每个像素的红、蓝、绿三色各有一个薄膜晶体管。当薄膜晶体管通导时,有电场加到液晶上,对光线进行调制。滤色片把可见光分解成三原色,进而组成各种颜色以得到彩色画面。

(3) 液晶显示器的特点　①液晶显示器的工作电压低,功耗低。②液晶显示器件的结构便于利用集成电路工艺进行批量生产。③无辐射。液晶显示器件不用高电压,在使用时不会产生 X 射线和电磁波辐射。④每个像素都有三个独立的晶体管,允许对每个像素直接寻址,并有记忆的功能。不需要扫描,也就没有闪烁和图像畸变。⑤重量轻、厚度薄、

体积远小于 CRT 显示器。

和 CRT 显示器比较，液晶显示器也有不足之处：①显示色域不够宽，颜色重现不够逼真；②响应速度偏低。液晶分子扭转过程需要较长的时间，当图像快速运动时会出现拖尾、模糊、残影等现象；③长时间使用可能产生亮点、暗点，寿命不及 CRT 显示器。④对比度差，可视角度小。

液晶显示器的控制与 CRT 显示器不同，要用相应配套的液晶适配器。

液晶显示器在电子表、仪器仪表和笔记本计算机等领域中已广泛应用。随着性能的提高和成本的降低，在台式计算机中的应用也越来越多。

等离子（PDP）显示器

等离子的发光原理是在真空玻璃管中注入惰性气体（氦、氖或氩），加电压之后，使气体产生等离子效应，产生的紫外光激发附近的涂布在玻璃上的荧光质，进而产生所需要的红、绿、蓝三原色。通过控制紫外光的强度来产生不同的亮度的三原色，组成各式各样的颜色。利用激发时间的长短来产生不同的亮度。这点跟 CRT 显示器一样，属于自体发光，因此它的亮度、颜色鲜艳度与屏幕反应速度，都跟 CRT 显示器相近。

等离子显示屏是由许多个放电小空间排列而成，每一个放电小空间称为单元。而每一个单元只控制红绿蓝三色中的一色，每一个像素由三个不同颜色（三原色）的等离子发光体单元组成。不同比例的三原色混合成像素的颜色。这点和液晶显示屏是相近的。

等离子显示器的特点是：①高对比度和亮度　PDP 显示器由一个个发光单元组成，不存在 CRT 显示器的模糊、闪烁和三原色不集中等问题。PDP 显示器的亮度可达 1 000 cd/m^2（坎每平方米）。对比度可达到 10 000∶1，可制造出全黑效果。②低辐射　等离子的电磁辐射只有 CRT 显示器的 1/100～1/1 000。③可作大面积显示屏　因为等离子显示屏的每个像素都能自己发光，每个像素的反应时间短，色彩饱和度高，适合制作大面积的显示屏，最大对角可达 381 cm。④厚度小　厚度可做到 6 cm。⑤等离子显示器是一种平面显示屏，且没有液晶显示器的可视角限制。⑥和液晶显示器比较，工作温度高，功耗大。

参考文献

1. 田民波，叶锋. TFT 液晶显示器原理与技术. 北京：科学出版社，2010

2. 周怡聪. 多媒体计算机外部设备. 北京：清华大学出版社，2002

（林兼　黄建忠）

xianshi bingxing zhiling jisuan

显式并行指令计算（explicitly parallel instruction computing，EPIC）　一种计算机系统结构设计思想。指令执行模式主要由编译器显式指定　系统结构为开发指令级并行性提供相应的硬件支持，并提供编译器与机器硬件之间的通信机制，满足以上两个条件的处理器称为采用 EPIC 思想设计的处理器。

显式并行指令计算是继**复杂指令集计算机**（complex instruction set computer，CISC）和**精简指令集计算机**（reduced instruction set computer，RISC）之后的又一种计算机系统结构设计思想。在采用 CISC 思想和 RISC 思想设计的处理机中，进入到处理机中执行的目标代码是串行的，每条指令中通常只包含一个操作。在程序执行过程中，由处理机内部的硬件对目标代码进行数据相关性分析、控制相关性分析和功能部件冲突检测，并进行并行调度，把没有数据相关性、控制相关性和功能部件冲突的多条指令调度到同时执行。进入 20 世纪末期，普遍采用的多发射、超流水线和乱序执行等技术，使一个处理机内同时执行的指令数量达到 100 条以上。对如此多的指令进行数据相关性分析和并行调度需要花费大量的硬件，但是能够达到的指令级并行度仍然不够高。

EPIC 设计思想的本质是重新划分处理机的软件与硬件功能，把需要花费大量不规则硬件的数据相关性分析、控制相关性分析、功能部件冲突检测和并行优化调度等工作由软件做，硬件对软件的优化工作提供必要的支持。显式并行指令计算的主要特点如下：

（1）指令执行模式由编译器显式指定。图 1 是安腾（Itanium）处理机的指令格式及一段程序代码。该处理机由 Intel 公司与 HP 公司在 20 世纪 90 年代中期联合研制，是一种典型的采用 EPIC 思想设计的处理器。

图 1（a）是安腾处理机的一段汇编代码，每个大括号内有 3 条指令，称为一个 bundle，字长为二进制 128 位。每个时钟周期可以同时发射两个 bundle，即 6 条指令。大括号内第一行的 3 个字母分别指明这 3 条指令的类型，m 为访存指令，i 为整数指令，b 为分支指令，f 为浮点指令。两对双分号之间的所

```
{ .mii
    add r1 = r2,r3
    sub r4 = r4,r5 ;;
    shr r7 = r4,r12 ;;
}
{ .mmi
    ld8 r2 = [rl] ;;
    st8 [rl] = r23
    tbit p1,p2 = r4,5
}
{ .mbb
    ld8 r45 = [r55]
(p3) br.call b1=funcl
(p4) br.cond Label1
}
{ .mfi
    st4 [r45] = r6
    fmac f1 = f2,f3
    add  r3 = r3,8 ;;
}
```

(a) 安腾处理机的一段程序

127–87	86–46	45–5	4–0
instruction slot 2	instruction slot 1	instruction slot 0	template
41	41	41	5

(b) 一个bundle的格式

OpCode	Reg1	Reg2	Reg3	Predicate
14	7	7	7	6

(c) 一条指令格式

图 1 安腾处理机的指令与程序

有指令可以同时被发射，它们之间没有数据相关、控制相关和功能部件冲突。

图 1(b)是一个 bundle 的格式，它包含有 3 条指令和一个 template 字段，每条指令长 41 位，template 字段长 5 位，指示这 3 条指令的类型及 stop(双分号)的位置。

(2) 硬件对编译器开发指令级并行性提供必要的支持　在采用 EPIC 思想设计的处理器中，通过硬件与软件的协同工作来提高程序的指令级并行度。硬件对编译器的支持主要包括以下几方面：①软件与硬件协同实现内存地址猜测执行。在高并行度处理机中，LOAD 操作是引起指令流水线停顿的主要原因。对于大量模糊相关的内存地址，编译器把这些 LOAD 操作提前执行，甚至越过 STORE 操作，这样可大幅度提高程序的执行速度。同时，硬件把 LOAD 操作的地址和该操作的位置等信息记录下来，如果猜测失败，执行编译器生成的恢复代码。②每条指令都有一个 Predicate 字段，如图 1(c)所示。该字段指示本条指令的启动时刻，用于对软件流水的装入代码与排空代码提供支持。

(3) 采用显式并行指令计算思想设计的处理器　其指令级并行度通常高于采用 RISC 思想和 CISC 思想设计的处理器。由于主要依靠编译器开发指令级并行度，同时有硬件的强有力支持，可以在一个循环、一个函数，甚至整个程序中寻找指令级并行性，可以采用软件流水、循环展开和并行优化调度等指令级并行度高的方法充分开发程序中的多种并行性。　(汤志忠)

xianchang kebiancheng menzhenlie

现场可编程门阵列(field programmable logic array, FPGA)　一种含有可编程元件、可供使用者在应用现场通过编程定制其功能的逻辑门阵列器件。FPGA 作为一种通用型器件的出现和迅速发展，改变了采用固定功能器件、自下而上的传统数字系统设计方法。使用 FPGA，用户可通过编程的方式实现所需逻辑功能，而不必依赖由芯片制造商设计和制造的集成电路芯片。

1990 年以来，FPGA 产品迅速发展并得到了广泛的应用。2000 年后，Altera 公司先后推出了带有嵌入式微处理器和嵌入式数字信号处理器的 FPGA 产品，使 FPGA 进入了可编程片上系统(SOPC)时代。

硬件结构

FPGA 的基本组成部分包括：可编程逻辑资源(CLB)、可编程互连资源(PI)，按照组(Bank)形式分布的可编程输入输出单元、集成知识产权(IP)等，其结构如图 1 所示。其中，可编程逻辑资源实现设计所需要的逻辑功能；可编程互连资源位于 CLB 阵列之间，通过编程配置实现逻辑单元之间以及逻辑单元与可编程输入输出模块的连接；可编程输入输出模块位于器件的四周，提供内部逻辑阵列与外部引出线之间的可编程接口。

可编程逻辑资源　主要功能是提供基本的逻辑运算操作和数据存储。每个 CLB 由多个基本逻辑单元(LE)组成。LE 通常由可编程的组合逻辑模块与可编程的寄存器逻辑(常用 D 触发器实现)相连接而构成。组合逻辑模块主要采用查找表(LUT)结构或多路选择器(MUX)结构：①基于 LUT 结构的逻辑模块主要应用于静态随机存储器(SRAM)存储结构的 FPGA 中，k 个输入信号的逻辑真值表通过编程保存在 $2^k \times 1$ 的 SRAM 中。输入信号作为这个 LUT 的地址信号。该结构可实现 k 个输入的任意组合逻辑功能。②基于 MUX 结构的逻辑模块主要应用于反熔丝和 Flash 存储结构的 FPGA 中。它利用多路选择器，通过对输入信号设置常量或变量，实现输入变量的多种组合功能。该结构可以使用较少的晶体管来实现较多的功能，但不能实现输入的所有可能的功能，从而对映射工具提出了很高的要求。此外，基本逻辑单元中通常还包含进位逻辑、存储资源等。

可编程互连资源　分布在各可编程逻辑功能块之间、逻辑功能块和可编程输入输出模块之间，为它们提供信号通路。可编程互连资源可以通过不同的结构实现，如通道型互连结构、层次化互连结构和孤岛型互连结构等。其中，孤岛型互连结构是应用最为广泛的互连方式，绝大部分基于 SRAM 存储单元的 FPGA 器件都采用了该结构。孤岛式体系架构的可编程逻辑阵列中的互连资源如图 2 所示，包括互连通道、逻辑功能单元的输入输出连接模块和交叉开关模块。

互连通道是由一定数量的规则连线单元所构成的。逻辑功能块之间的信号传输绝大多数是通过这

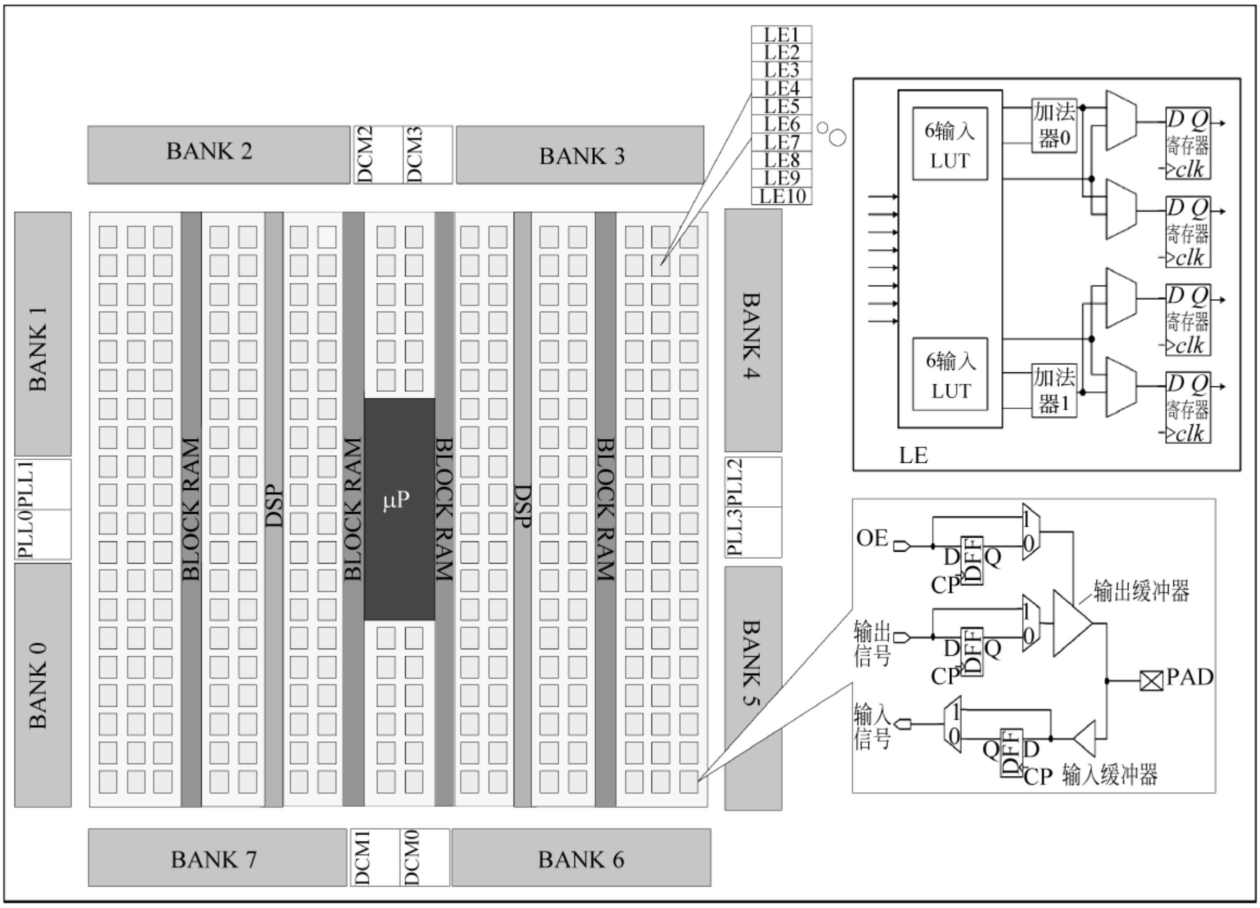


图 1　FPGA 结构

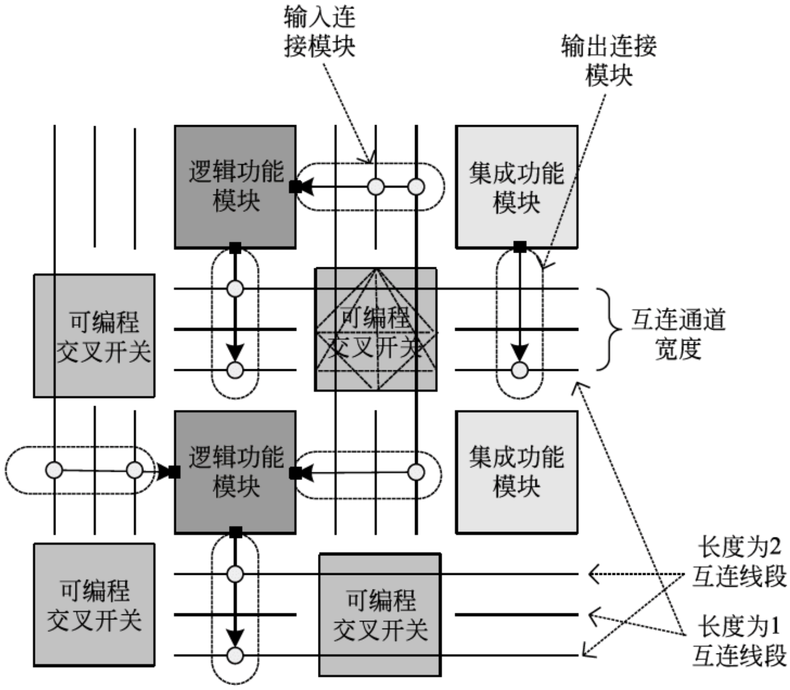


图 2　孤岛式体系架构下可编程互连资源

些连线单元来实现的,包括长互连线段和短互连线段。其中,短互连线段用于实现相邻逻辑功能块之间的互连,长互连线段用于满足相距较远的逻辑功能块之间的信号通路需求。互连通道的数目、方向、长度、组成比例和分布,影响了电路的延时特性和信号质量,也直接决定了可编程逻辑阵列的布通率和资源利用率。

输入输出连接模块实现逻辑功能模块和连线通道之间信号的传输,与逻辑功能块四周的端口数目和信号传递需求相关。

互连资源有着多种电路实现形式,通常是通过传输门、缓冲器、多路选择器等基本电路单元来实现的。通过 SRAM 控制这些基本电路的导通或关断以实现信号传递的可编程。

可编程输入输出模块　位于器件的四周,提供了 FPGA 与片外电路的接口。FPGA 的通用性和可编程性决定了其应用领域的多样性。与按应用需求定制的集成电路(ASIC)中的输入输出模块不同,FPGA 中的输入输出模块需要满足不同的速度与电平标准。对同一个用户输入输出模块,用户通过编程,可以实现输入、输出、双向以及寄存后输入输出等多种功能,同时可以根据用户的要求,对其电流驱动能力、摆率以及输入输出信号的延时进行控制。目前主流 FPGA 大多支持多种 I/O 标准,用户可根据自己的需求以及芯片供应商提供的设计方案选择合适的信号标准。

可编程知识产权(IP)资源　随着半导体工艺技术的不断发展,用户对 FPGA 实现电路的复杂度和性能也有着更高的需求,FPGA 在自身密度和速度不断提高的同时,也集成了越来越多的功能模块。如:

(1) 通常采用硬核的方式嵌入可编程存储器(BLOCK RAM),用户可以通过配置的方式实现存储器不同的存储方式和工作模式,如单端口、双端口随机存取存储器,以及先进先出(FIFO)等。

(2) 通过集成软核或硬核微处理器,采用多核并行处理技术,实现设计复用,降低 CPU 的功耗和存储资源,节约系统成本。通过集成高性能数字信号处理快速经济完成设计,实现各种复杂度的数字信号处理算法。

(3) 通过集成锁相环或时钟管理器为 FPGA 产生种类丰富的时钟资源,支持倍频、分频、相位移动、占空比调节等功能。

(4) 将嵌入式高速收发器作为独立专用电路模块集成在 FPGA 中,以适用通信总线协议与接口标准,满足接口高速数据传输需求。

(5) 常以软核的形式实现嵌入式逻辑分析仪,无须进行任何外部探测或修改便可获取设计中任意的内部节点或输入输出引脚的状态,以零成本和系统级的速度实时捕获和显示 FPGA 中的信号,对系统进行观测和调试。

设计与编程

FPGA 设计是 EDA(参见**电子设计自动化**)的一种特殊形式。FPGA 设计软件给 FPGA 用户提供了一个应用设计开发平台,设计者在此平台上,用硬件描述语言 HDL(Verilog、VHDL 等)或原理图完成具体设计。

FPGA 设计的流程依次包括设计输入、综合、工艺映射、装箱、布局、布线、时序验证和位流码生成和下载等阶段。其中,不同于普通 EDA 设计,较有 FPGA 特点的阶段有:①工艺映射是把与工艺无关的数字电路描述转化为与工艺相关、功能等价的电路描述;②装箱是指将逻辑综合后产生的基本逻辑单元组装到逻辑块或宏单元中,装箱的目标是提高逻辑块的利用率,尽可能减少逻辑块之间的外部信号连接,以达到减少电路面积、提高电路性能的目的;③位流码生成是根据布局布线结果将电路转化成由 0 和 1 组成的数据文件的过程,位流码下载是通过一些配置方式将位流码下载到 FPGA 芯片中,实现用户电路功能的过程。

随着 FPGA 从一种复杂的可编程逻辑器件向可编程**片上系统**方向发展,FPGA 器件也从通用型半导体器件向平台化的系统级器件发展。越来越多的工艺、器件和电路新技术被 FPGA 研究者采用,可能在未来对 FPGA 设计产生重大影响。例如,3D 集成技术可利用 FPGA 自身重复性结构的特点并有助于缓解其互连延时问题;采用无源忆阻器(参见**非易失随机存取存储器**)替代 SRAM 存储单元,能够有效降低 FPGA 芯片中晶体管的数量和功耗,提高器件的密度,同时具有可重构性和非易失性;采用异步电路技术进一步提高 FPGA 的性能,等等。

参考文献

1. Betz V. 深亚微米 FPGA 结构与 CAD 设计. 王伶俐,杨萌,周学功,译. 北京:电子工业出版社,2008

2. Maxfield C. FPGA 设计指南:器件、工具和流程. 杜生海,邢闻,译. 北京:人民邮电出版社,2007

(杨海钢)

xianchang zongxian kongzhi xitong

现场总线控制系统(fieldbus control system, FCS)　基于现场总线的控制系统。它通过现场总线把多个测量控制仪表、计算机等作为结点连接成网络系统,通过公开、规范、标准的通信协议,在位于生产控制现场的多个微机化自控设备之间以及现场仪表与用作监控、管理的远程计算机之间,实现数据传输与信息共享。这里的现场总线是指应用在生产现场,在微机化测量控制设备之间实现双向串行多结点数字通信的底层局部控制网络。

系统组成　与传统**集散控制系统** DCS 类似,现场总线控制系统由现场总线智能仪表、现场总线网络设备、传输介质、现场总线控制器以及操作、监控与组态计算机工作站组成。

现场总线智能仪表是指具有现场总线通信接口的一次仪表、二次仪表,除具有传统模拟仪表的测量、执行功能外,还具有智能化的信息处理与现场总线通信功能。有的现场总线仪表还内置有软件功能块(function block,FB),如模拟量输入块(AI)、模拟量输出块(AO)、数字量输入块(DI)、数字量输出块(DO)等,有的现场仪表还内置有控制功能块,如 PID 功能块。

现场总线网络设备是指现场总线连接器、中继器、网桥、终端器、**网关**、链接设备、现场总线耦合器以及符合现场总线物理层特性的安全栅、总线电源等设备。如图 1 所示。其中,连接器(connector),亦称作接插件、插头和插座,用于连接两个现场总线设备的器件。中继器(repeater)是网络物理层上面的连接设备,适用于完全相同的两类网络互联,主要功能是通过对数据信号的重新发送或者转发,来扩大网络传输的距离。网桥(bridge)是一种在链路层实现中继的设备,常用于连接两个或更多个现场总线网段的网络互联设备。终端器(terminator)安装于总线的两端,主要有两个作用:一是对总线特性阻抗的匹配;二是实现总线调制电流到总线电压信号的转换。网关(gateway)是在采用不同现场总线协议的网络之间进行互通时,用于提供协议转换、路由选择、数据交换等网络兼容功能的设施。链接设备(link device)是指用于连接一个现场总线体系中高、低速不同网段(如 FF 的 HSE 与 H1)间的连接设备。总线耦合器(bus coupler)是连接一个现场总线体系中不同类型总线网段(如 Profibus 的 DP 与 PA)间的连接与转换设备。而现场总线安全栅、总线电源等设备则在物理特性上符合工程应用中所使用的现场总线物理信号定义。其中,链接设备、现场总线耦合器、网桥与网关还具有网段分隔与连接的功能。

传输介质包括双绞线、**光缆**、**同轴电缆**以及无线传输介质等。

现场总线控制器指具有现场总线接口和通信功

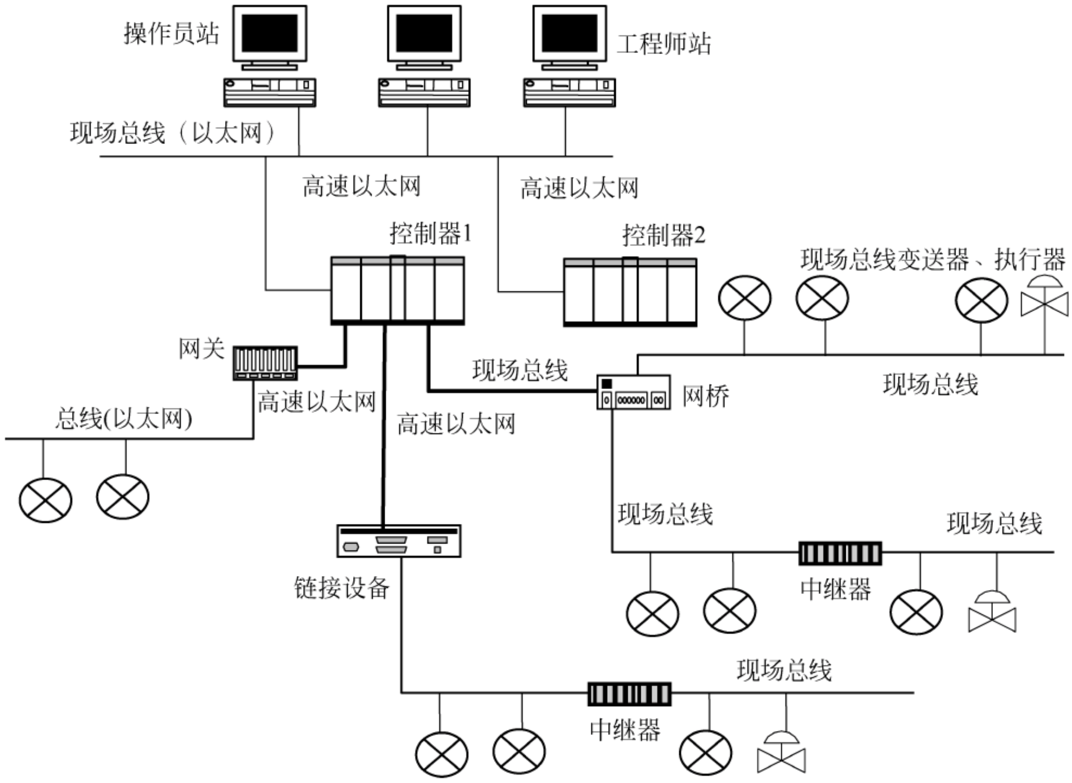


图 1 现场总线控制系统结构示意图

能的控制器。

操作、监控与组态计算机工作站与传统 DCS 类似,其中组态软件除了对测量、控制策略进行组态外,还需要对现场总线仪表的网络特性参数(如地址、端口、通信链路关系、通信角色等)进行配置;对有些具有功能块 FB 的仪表,还需要进行功能块参数的配置。

有的现场总线控制系统还配置有设备管理软件(或资产管理软件),其功能是利用获得的现场总线设备相关信息(如制造商、设备类型等)、诊断信息、管理信息等,对现场总线设备进行组态管理、远程故障诊断、远程调试与标定以及对设备数据进行自动维护等。

现场总线特点 与传统控制系统相比,现场总线控制系统(FCS)有如下优点:

(1) 全数字化 将企业管理与生产自动化有机结合一直是工业界梦寐以求的理想,但只有在 FCS 出现以后这种理想才有可能高效、低成本地实现。在采用 FCS 的企业中,用于生产管理的局域网能够与用于自动控制的现场总线网络紧密衔接。此外,数字化信号固有的高精度、抗干扰特性也能提高控制系统的可靠性。

(2) 可实现分布式测量与控制 在 FCS 中各现场设备有足够的自主性,它们彼此之间相互通信,完全可以把各种控制功能分散到各种设备中,而不再需要一个中央控制计算机,实现真正的分布式控制。

(3) 双向的数据传输 传统的 4 ~ 20 mA 电流信号,一条线只能传递一路信号。现场总线设备则在一条线上既可以向上传递传感器信号,也可以向下传递控制信息。

(4) 自诊断 现场总线仪表本身具有自诊断功能,而且这种诊断信息可以送到中央控制室,以便于维护,而在只能传递一路信号的传统仪表中是做不到的。

(5) 节省布线及控制室空间 传统的控制系统每个仪表都需要一条线连到中央控制室,在中央控制室装备一个大配线架。而在 FCS 系统中多台现场设备可串行连接在一条总线上,这样只需较少的线进入中央控制室,大量节省了布线费用,同时也降低了中央控制室的造价。

(6) 仪表功能的多重化 数字、双向传输方式使得现场总线仪表可以摆脱传统仪表功能单一的制约,可以在一个仪表中集成多种功能,做成多变量变

送器,甚至集检测、运算、控制于一体的变送控制器。

(7) 开放性　现场总线不再是专有的协议,而是通过国际标准(如 IEC 61158)、地区标准、国家标准、行业标准发布的公开、开放的协议。

(8) 互操作性　来自不同厂家、遵循同一协议的现场总线设备可以互操作,这样就可以在一个企业中由用户根据产品的性能、价格选用不同厂商的产品,集成在一起,避免了传统控制系统中必须选用同一厂家的产品限制,促进了有效的竞争,降低了控制系统的成本。

(9) 智能化与自治性　现场总线设备能处理各种参数、运行状态信息及故障信息,具有很高的智能,能在部件甚至网络故障的情况下独立工作,大大提高了整个控制系统的可靠性。

现场总线标准　现场总线技术自 20 世纪 90 年代初开始发展以来,一直是世界各国关注和发展的热点,目前具有一定规模的现场总线已有数十种之多,为了开发应用以及争夺市场的需要,世界各国纷纷制订了各自的国家标准(或协会标准),同时力求将自己的协议标准转化成各区域标准化组织的标准。

目前现场总线标准由国际电工委员会 IEC/TC65/SC65C/MT9 负责制定与修订。现场总线技术在历经“群雄并起”“分散割据”的初始阶段后,经历了十多年的纷争,2000 年形成了一个由 8 个类型组成的 IEC61158 现场总线国际标准。该标准于 2003 年启动修订、于 2007 年底发布的第 2 版本,更是包括了 16 大类、20 个类型,成为国际上制订时间最长、技术类型最多、包容度最大的国际标准之一(见表 1)。

表 1　现场总线协议类型

类型	技术名称
Type1	61158-1 现场总线
Type2	CIP 现场总线
Type3	Profibus 现场总线
Type4	P-Net 现场总线
Type5	FF-HSE 高速以太网
Type6	SwiftNet(被撤销)
Type7	WorldFIP 现场总线
Type8	Interbus 现场总线
Type9	FFH1 现场总线
Type10	ProfiNet 实时以太网
Type11	TCnet 实时以太网
Type12	EtherCAT 实时以太网
Type13	Powerlink 实时以太网
Type14	EPA 实时以太网
Type15	Modbus-RTPS 实时以太网
Type16	SERCOS-Ⅰ、Ⅱ现场总线
Type17	VNET/IP 实时以太网
Type18	CC-Link 现场总线
Type19	SERCOS-Ⅲ实时以太网
Type20	HART 现场总线

工业实时以太网　工业实时以太网技术是 2000 年左右迅速发展起来的新型现场总线技术,它利用了成本低、稳定性好和可靠性高、应用广泛、共享资源丰富、易于与 Internet 连接等特点,以普通以太网技术(ISO/IEC8802.3)为基础,进行了某些特性和协议的改良,以满足工业控制网络的通信确定性、实时性等特殊需求。因此工业以太网技术是普通以太网技术在工业控制网络延伸的产物,也符合现场总线由低速向高速发展的趋势。

2003 年,IEC/SC65C 正式决定制定工业以太网国际标准; 2007 年 12 月,IEC 发布了现场总线国际标准 IEC 61158(第二版),收录了包括中国浙江大学、中控集团等主持制定的 EPA(Ethernet for Plant Automation)、德国 BECKHOFF 公司的 EtherCAT、日本横河的 V-net、日本东芝的 TCnet、奥地利 B&R 公司的 PowerLink、法国施耐德的 MODBUS/TCP(RTPS)等在内的工业实时以太网协议。这些实时以太网技术与传统现场总线一起,组成了 IEC61158 新的现场总线国际标准体系,并均作为实时以太网应用行规国际标准 IEC61784-2 的子集。这样,IEC 61158 中包含的现场总线(包括传统现场总线和实时以太网)类型由原来的 11 种扩展到了 20 种(包括 10 种实时以太网技术)。

工业无线通信　工业无线通信技术是最近几年迅速发展起来的新型控制网络技术。由于无线通信的诸多优势,也随之推动了无线通信技术在工业自动化领域的应用。无线通信技术将超越地域和空间的限制,在某些远程化、移动对象等应用场合具有较强的优势,但总体上讲,由于在安全性、信号干扰等方面的缺点,工业无线通信技术目前还是有线通信

技术的一种补充。

应用于工业控制网络的无线通信技术,可分为远程无线通信技术和短程无线通信技术,其中远程无线通信技术包括无线电台远传技术、GSM 远传技术、GPRS(CDMA)远传技术、3G 远传技术等,而短程无线通信技术包括 IEEE 802.11、IEEE 802.15、IEEE 802.15.4 等。

在短程无线通信技术中,IEEE 802.15.4 协议受到了自动化领域的广泛关注,特别是由美国仪器仪表、系统与自动化协会 ISA 制定的 ISA-100 和美国 HART 基金会制定的 WirelessHART™最具代表性和竞争性。

参考文献

1. IEC (International Electrotechnical Commission, 国际电工委员). IEC 61158 (all parts) Industrial communication networks-Fieldbus specifications, 2007

2. IEC (International Electrotechnical Commission, 国际电工委员). IEC 61784-2 Industrial communication networks-Profiles-Part 2: Additional fieldbus profiles for real-time networks based on ISO/IEC 8802-3

3. 冯冬芹,黄文君. 工业通信网络与系统集成. 北京:科学出版社,2005 (冯冬芹)

xiandai fuwuye

现代服务业(modern service industry) 指依靠高新技术(特别是信息技术)和现代管理方法、经营方式及组织形式发展起来的、信息和知识密集型的、高附加值的服务业。它既包括随着信息网络技术发展形成的新兴服务业,也包括传统服务业经技术改造升级和经营模式更新而形成的新型服务业。其本质是实现服务业的现代化:例如,金融服务、**电子商务**服务、**电子政务**服务、信息技术与网络通信服务、现代教育培训服务、现代物流服务等服务业,以及各种被高新技术改造过的制造产品服务、旅游服务、社会服务等传统服务业。

现代服务业是由我国提出的相对于传统服务业的新概念。相对于传统服务业,现代服务业一般具有五大基本特性:

(1) 高技术性 现代服务业科技含量高,特别是信息技术含量高。

(2) 知识性 现代服务业是知识密集型产业,它为消费者提供知识的生产、传播和使用服务,使知识在服务过程中实现增值。

(3) 高增加值性和集群性 现代服务业不仅可以使服务过程产生知识的增值,而且可以产生服务的规模效应和各种服务相互融合的聚集效应,引起服务的大幅度增值,为顾客带来高价值。

(4) 从业人员高素质性 现代服务业的从业人员大都具有良好的教育背景、专业知识基础和技术、管理的能力,构成现代服务业的核心能力。

(5) 新兴性 现代服务业是新兴服务业或从过去传统服务业升级演变而来的新型服务业。技术和知识的广泛应用引发了新业务模式,使服务价值交换关系、服务资源配置模式、服务运作过程发生深刻改变,强调专业化分工和服务创新。

同时,现代服务业还具有以下的时代特征:①新服务领域 现代服务业适应现代城市和现代产业的发展需求,突破了消费性服务业的领域,形成了新的生产性服务业、智力(知识)型服务业和公共服务业的新领域。②新服务模式 现代服务业通过服务功能换代和服务模式创新,而产生了新的服务业态。③现代服务业具有高文化品位和高技术含量;高增值服务;高素质、高智力的人力资源结构;高感情体验、高精神享受的消费服务质量。

因此,现代服务业可以用如下等式加以表示:

现代服务业 = 新兴服务业 + 传统服务业 + 新业务模式 + 新型 IT 技术 + 知识密集 + 高增值性

现代服务业可划分为四类:

(1) 基础服务 包括通信服务和信息服务;

(2) 生产和市场服务 主要包括金融、物流、批发、电子商务、农业支撑服务、中介和咨询等专业服务;

(3) 个人消费服务 主要包括住宿、餐饮、房地产、文化娱乐、旅游、商品零售等;

(4) 公共服务 主要包括政府公共管理服务、教育、科技、公共卫生与医疗保健以及公益性信息服务等。

传统服务业主要通过以下四种典型手段及其复合而发展成为现代服务业:

(1) 服务外包(service outsourcing) 企业在经营业务中仅保留自己少数具有核心竞争力的运行功能,将那些自己不擅长的服务业务外包给专业化服务公司,通过在外部市场购买专业化服务来降低成本,提高效率和效益。服务外包广泛存在于各类服务业当中,例如数据处理外包、呼叫中心外包、IT 服务外包、物流外包等。

(2) 服务聚合(service mashup) 把由不同提

供者提供的多项独立分散的服务按照特定的逻辑整合在一起，形成新的大粒度服务并向外提供，创造出全新的服务价值。服务聚合的典型示例有：携程旅行网（将旅行社、航空公司、酒店、银行、保险公司、电信运营商等提供的基础性服务整合在一起）、阿里巴巴/淘宝网（将大量的卖家和买家聚集在统一的电子商务平台上，支持双方之间的电子商务交易）、第四方物流（将各物流企业的分布式资源整合起来，提供门到门的物流运输服务）等。

（3）服务虚拟化（everything as a service，EaaS）在IT技术和**互联网**的支持下，对现实中存在的服务进行虚拟化，使之可以通过互联网的渠道突破时空局限向外发布，并更加容易的被顾客所访问、获取和使用。服务虚拟化分为两大类。一类是对计算资源的虚拟化和服务化，例如软件即服务（SaaS）、**云计算**等。另一类是对传统行业的服务化，例如R&R公司的TotalCare服务（不直接出售发动机，而以“租用服务时间”的形式出售，并承担一切保养、维修等服务）等。

（4）社会化、本地化、移动化（social，local，mobile，简称SoLoMo） 社会化是将现实中人与人之间的社交网络映射到互联网虚拟空间中，进而支持各类互联网应用服务；本地化是指通过无线网络或外部定位方式获取移动终端用户的位置信息，进而提供各类与位置相关的增值服务应用；移动化是指借助各类移动终端接入服务系统，随时随地访问云端服务。典型SoLoMo的代表有社交网络服务（Facebook、Twitter等）、基于位置的服务（LBS）、云手机、移动电子商务等。

现代服务业需要多个技术学科综合交叉，共同支撑，其中，信息技术和现代管理科学是支柱性的技术基础。①以计算机和网络通信为核心的信息科学技术催化了现代服务业的形成与不断创新。这方面的典型支撑技术包括：新型互联网技术、电子商务、**Web 2.0**、网络搜索引擎、RFID（无线射频标识技术）、**物联网**、务联网、信息安全技术、云计算技术及环境等。②现代管理科学引领了服务业从经验管理向科学管理、由面向生产者向面向顾客的服务理念创新。服务管理创新理论主要包括：现代服务模式创新、服务市场创新、服务过程创新、人力资源管理创新、服务组织系统创新、服务创新测度和绩效评价等。③为现代服务业提供理论和技术支持的还有经济学、工程学、数学、人文社会科学等其他相关科学。

现代服务业的主要发展趋势为：①现代服务业在全球范围内趋于持续快速增长趋势，并成为各国家（尤其是发达国家）经济发展的支柱性产业；②新业务模式是现代服务业的重要特征，服务模式创新日益成为现代服务业竞争的核心；③随着服务外包成为现代服务业国际化迁移的重要途径，基于网络的第三方服务模式将成为现代服务业的主流模式；④随着互联网和云计算及其资源的不断丰富，信息资源、信息技术及信息网络运行平台将成为现代服务业的主导要素；⑤随着现代服务业的分化与融合，不断变化的垄断竞争市场将成为现代服务业主导性市场结构；⑥随着工业化和信息化融合的发展进程，信息技术服务化、制造业服务化、服务业信息化趋势日益明显，现代服务产业与传统产业相互作用与融合越来越深入。 （徐晓飞 王忠杰）

xianding luoji

限定逻辑（circumscription logic） 通过对给定的背景理论中的谓词加以限定的方法处理常识推理的一种**非单调逻辑**。John McCarthy为常识推理的形式化而提出的限定逻辑是非单调逻辑中最老，最有影响，研究得最为透彻的一种。在常识推理中，人们通常并不对一条规则的所有可能的例外情况一一细察。McCarthy是最早观察到这一点的人之一。例如，如果你知道Fido是一条狗，你就会得出“Fido会叫”的推论，而不会去对诸如“Fido是一条刚刚出生的狗，还不会叫”这样的例外一一考虑。在最简单的形式下，限定逻辑对给定的背景理论T中的谓词P加以限定，即只考虑T的包含着满足性质P的最小个体集合的模型。在Fido的例子中，我们可以在背景理论中引入一个谓词“abnormal”，并增加下述规则：

（1）for all x, if x is a dog and x is not abnormal, then x barks

如果知道如下事实：

（2）Fido is a dog

我们并不能从1和2直接推出“Fido会叫”，因为Fido有可能是一只abnormal的狗。这时候我们就需要对1和2中的谓词abnormal进行限定。该限定逻辑的模型是1和2的包含着满足性质abnormal的最小个体集合的模型。此例中的最小个体集合就是空集合，即没有满足性质abnormal的个体。于是，在该限定逻辑的所有模型中，Fido都不是abnormal的，所以是会叫的。当然此例过于简单，在实际问题中，如何施行限定不是个简单的问题。例如，我们可

以说,通常鸟会飞并且长羽毛。为了使用限定,我们需要两个表达非正常情况的谓词,一个用于会飞,一个用于长羽毛:

if x is a bird and x is not abnormal-fly, then x flies

if x is a bird and x is not abnormal-feather,
 then x has feathers

我们又知道通常企鹅不会飞。所以我们还需要如下这类规则:

if x is a penguin, then x is a bird and x is abnormal-fly

if x is a penguin, and x is not abnormal-penguin-fly,
 then x does not fly

如果我们后来又发现了某种企鹅其实会飞,则需要增加一个新的表达非正常情况的谓词和相应的规则.可以想象这个过程将如何继续下去,并且理解我们为何需要系统的方法才能将限定用于常识推理的形式化。(感兴趣的读者可以阅读下面参考文献列出的 McCarthy 原始论文,以获得更详细的信息。)

形式化地说,限定逻辑是二阶逻辑。例如:一阶语句 W 中的谓词 $P(x)$ 的限定是如下的二阶语句:

$$W \ \& \ \text{forall}\ P' \ (W(P') \ \& \ P' \text{ in } P \Rightarrow P \text{ in } P')$$

此处 P'是谓词变量,$W(P')$ 是在 W 中将 P 换成 P'所得的结果,P' in P 表示 forall x $(P'(x) \Rightarrow P(x))$,而 P in P'具有类似的定义。

在某些特殊情况下,限定可以简化为一阶语句。最简单的例子是当背景理论 W 为一组如下的规则:

$$\text{forall}\ x \ (A_1 \Rightarrow P(x))$$
$$\cdots$$
$$\text{forall}\ x \ (A_k \Rightarrow P(x))$$

其中 $A_1,\cdots,A_k$为不含 P 的公式。此时,P 在 W 中限定等价于如下的语句:

$$\text{forall}\ x \ (P(x) \Leftrightarrow A_1 \vee \cdots \vee A_k)$$

这恰好是 Clark 提出的定义一阶逻辑的逻辑程序语义的 Clark's completion。

限定逻辑已在许多领域得到应用,如:商业的形式化合同语言,模型改变中的惯性形式化描述,自然语言理解,法律逻辑等。

参考文献

1. McCarthy J. Circumscription—A form of non-monotonic reasoning. Artificial Intelligence (1980), 13(1-2): 27-39

2. McCarthy J. Applications of circumscription to formalizing common-sense knowledge. Artificial Intelligence(1986), 28(1): 89-116

(林方真)

xiancheng

线程(thread) 并发程序(参见**进程**)中共享地址空间的并发执行单位。每个线程是一个并发执行单位,它拥有为执行计算所必须的程序代码及为**程序**运行所拥有的存储地址空间。一个**中央处理器**可以被多个线程并发地共享,而多个线程共享同一个地址空间可以为线程的分时运行和分时调度提供便利,从而使线程之间的切换所引起的系统开销减少。从并发程序的运行来看,线程是一种轻量级的进程。

线程的系统特性

早期,在**操作系统**中采用进程作为并发程序的并发执行单位。进程同时作为资源分配的基本单位和并发程序分时调度的基本单位。每个进程拥有各自独立的地址空间和为支持运行所拥有的其他资源。由于系统所拥有的地址空间和资源有限,从而限制了系统所能允许的最大的并发进程总数。同时,进程作为资源分配的单位也影响到进程切换和分时调度的系统开销,使进程的频繁切换消耗过多的处理器时间。20 世纪 80 年代引入了多线程进程技术,在一个进程中允许创建多个线程,从而把支持并发程序运行的两项任务——“动态分配资源”和“计算的分时调度”分离开来。前一项任务仍由进程来完成,它不需频繁切换。后一项任务则交给称为线程的并发执行单位来完成,允许处理器在多线程间较频繁地切换。线程有如下的一些特性:

(1) 每个线程有一个唯一的标识符和一个线程描述表,记录了线程执行的寄存器以及堆栈等现场状态。

(2) 一个进程中可包括若干线程,各个线程共享该进程的内存地址空间。

(3) 线程在创建后便开始了它的生存周期。在生存周期内会经历等待态、就绪态和运行态等状态变化,这种状态切换只涉及线程描述表内容。

(4) 进程中的线程可有多种组织方式。第一种是调度员-工作者模式,第二种是队列模式,第三种是有限状态机模式等。

现代的操作系统,如 Mach, Solaries, Windows 2000/XP 等都支持多线程进程和多线程程序设计。美国 IEEE 推出了针对 UNIX 类操作系统的多线程程序设计标准 POSIX1003.4a。

多线程程序设计与线程包

线程包是一套提供给用户编程使用的库调用原语集。有两种类型的线程包,分别是在用户空间中运行的线程包和内核中运行的线程包。**UNIX 操作系统**

虽不支持内核线程,但却提供用户空间线程包,例如POSIX 线程包。POSIX 规定了 UNIX 类操作系统(包括 Linux)线程库标准。传统的高级**程序设计语言**不提供多线程包,但 **Java 语言**为程序设计人员提供了一套 Thread 类,使得可以在 Java 程序中创建线程。

近年来,已经出现了有效支持多线程运行特征的微处理器体系结构,如 Intel 的奔腾 4 **微处理器**等。多处理器的芯片可以同时执行多个并发线程代码流。每个处理器都可独立地被中断或执行新的线程,而不影响芯片上其他处理器的工作。

参考文献

1. 陈向群,杨芙清. 操作系统教程. 北京:北京大学出版社,2001

2. 尤晋元,史美林,等. Windows 操作系统原理. 北京:机械工业出版社,2001

3. Tanenbaum A S. Modern operating systems. 2nd ed. Prentice Hall,2001

(陈向群)

xianxing daishu fangchengzu shuzhi jiefa

线性代数方程组数值解法(numerical solution for system of linear algebraic equations) 采用数值方法求线性代数方程组数值解的方法。通常,解线性方程组的数值方法包括直接法和迭代法两大类。凡经过有限次数运算可求得(如果没有舍入误差)方程组精确解的方法称为直接法。凡按照一定的迭代公式,从某一组初值出发,逐步逼近精确解的方法称为迭代法。对于阶数 n 固定的方程组,如果不考虑舍入误差,使用直接法能在预定的运算次数内求得精确解,然而原始数据的误差和运算的舍入误差是不可避免的,实际上只能获得方程组的近似解。对于中等规模($n<100$)的方程组及大型稀疏方程组,由于直接法所具有的准确性和可靠性,也经常使用它们来求解。对于极少量非零元素的大型稀疏线性方程组,特别是某些偏微分方程数值求解过程中出现的线性方程组,用直接法计算代价较高,迭代法更具竞争力。对于某些问题,迭代法可能发散或收敛很慢,应根据具体问题选用合适的迭代方法。

直接法

直接法中最基本的方法是消去法。重要的消去法有高斯消去法、列主元消去法、高斯-约当消去法和矩阵直接三角分解法等。

高斯消去法 考虑 n 阶线性方程组

$$\left.\begin{aligned} a_{11}^{(1)}x_1+a_{12}^{(1)}x_2+\cdots+a_{1n}^{(1)}x_n&=a_{1,n+1}^{(1)}\\ a_{21}^{(1)}x_1+a_{22}^{(1)}x_2+\cdots+a_{2n}^{(1)}x_n&=a_{2,n+1}^{(1)}\\ &\vdots\\ a_{n1}^{(1)}x_1+a_{n2}^{(1)}x_2+\cdots+a_{nn}^{(1)}x_n&=a_{n,n+1}^{(1)}\end{aligned}\right\}\qquad(1)$$

其中系数 a_{ij} 为实数或复数,方程组(1)的矩阵向量形式为

$$\boldsymbol{A}^{(1)}\boldsymbol{x}=\boldsymbol{b}^{(1)}\qquad(2)$$

其中

$$\boldsymbol{A}^{(1)}=(a_{ij}^{(1)}),\quad \boldsymbol{x}=(x_1,x_2,\cdots,x_n)^{\mathrm{T}}$$
$$\boldsymbol{b}^{(1)}=(a_{1,n+1}^{(1)},a_{2,n+1}^{(1)},\cdots,a_{n,n+1}^{(1)})^{\mathrm{T}}$$

$\boldsymbol{A}^{(1)}$ 与 $\boldsymbol{b}^{(1)}$ 为已知,$\boldsymbol{x}$ 为待求。

高斯消去法由消元过程和回代过程组成。消元过程是依次利用前面方程消去后面方程中的未知数,将方程组(1)变为等价的三角形方程组,然后通过回代过程求三角形方程组的解。

消元过程如下。设 $a_{11}^{(1)}\neq 0$,将(1)式中第 1 个方程分别乘以 $-a_{21}^{(1)}/a_{11}^{(1)}$, $-a_{31}^{(1)}/a_{11}^{(1)}$, $\cdots$, $-a_{n1}^{(1)}/a_{11}^{(1)}$ 加到第 2 至第 n 个方程上,消去这些方程中的 x_1,得到与(1)等价的方程组

$$\left.\begin{aligned} a_{11}^{(1)}x_1+a_{12}^{(1)}x_2+\cdots+a_{1n}^{(1)}x_n&=a_{1,n+1}^{(1)}\\ a_{22}^{(2)}x_2+\cdots+a_{2n}^{(2)}x_n&=a_{2,n+1}^{(2)}\\ \cdots&=\cdots\\ a_{n2}^{(2)}x_2+\cdots+a_{nn}^{(2)}x_n&=a_{n,n+1}^{(2)}\end{aligned}\right\}\qquad(3)$$

其中,后 $n-1$ 个方程的系数计算公式为

$$\left.\begin{aligned} l_{i1}=\frac{a_{i1}^{(1)}}{a_{11}^{(1)}},a_{ij}^{(2)}=a_{ij}^{(1)}-l_{i1}a_{1j}^{(1)}\\ i=2,\cdots,n;j=2,\cdots,n+1\end{aligned}\right\}\qquad(4)$$

方程组(3)的矩阵向量形式为

$$\boldsymbol{A}^{(2)}\boldsymbol{x}=\boldsymbol{b}^{(2)}$$

一般,设已得到第 $k-1$ 次消去后的结果

$$\boldsymbol{A}^{(k)}\boldsymbol{x}=\boldsymbol{b}^{(k)}\qquad(5)$$

则第 k 步消元过程如下:设 $a_{kk}^{(k)}\neq 0$,将式(5)中第 k 个方程分别乘以 $-a_{k+1,k}^{(k)}/a_{kk}^{(k)}$, $-a_{k+2,k}^{(k)}/a_{kk}^{(k)}$, $\cdots$, $-a_{nk}^{(k)}/a_{kk}^{(k)}$ 加到式(5)中第 $k+1$ 至第 n 个方程上,消去这些方程中的 x_k,得到与式(5)等价的方程组

$$\boldsymbol{A}^{(k+1)}\boldsymbol{x}=\boldsymbol{b}^{(k+1)}\qquad(6)$$

其中,后 $n-k$ 个方程的系数计算公式为

$$\left.\begin{aligned} l_{ik}=a_{ik}^{(k)}/a_{kk}^{(k)},a_{ij}^{(k+1)}=a_{ij}^{(k)}-l_{ik}a_{kj}^{(k)},\\ i=k+1,\cdots,n;j=k+1,\cdots,n+1\end{aligned}\right\}$$

如果依次 $a_{kk}^{(k)}\neq 0$, $k=1,2,\cdots,n-1$,则上述过程重复执行 $n-1$ 次后得到与式(1)等价的三角形方程组

$$A^{(n)}x = b^{(n)} \tag{7}$$

具体形式是

$$\left.\begin{aligned} a_{11}^{(1)}x_1 + a_{12}^{(1)}x_2 + \cdots + a_{1n}^{(1)}x_n &= a_{1,n+1}^{(1)} \\ a_{22}^{(2)}x_2 + \cdots + a_{2n}^{(2)}x_n &= a_{2,n+1}^{(2)} \\ &\vdots \\ a_{nn}^{(n)}x_n &= a_{n,n+1}^{(n)} \end{aligned}\right\} \tag{8}$$

以上对 k 的递推过程完成后，就完成了整个消元过程。如果 $a_{nn}^{(n)} \neq 0$，则由式(8)得出向后递推公式

$$\left.\begin{aligned} x_n &= a_{n,n+1}^{(n)}/a_{nn}^{(n)}, \\ x_i &= \Big(a_{i,n+1}^{(i)} - \sum_{j=i+1}^{n} a_{ij}^{(i)}x_j\Big)\Big/a_{ii}^{(i)}, \\ i &= n-1, n-2, \cdots, 1 \end{aligned}\right\} \tag{9}$$

称为回代过程。

用高斯消去法求解 n 阶方程组时，其乘除法和加减法的运算量约为$\frac{1}{3}n^3 + O(n^2)$。

在消元过程中有可能遇到某个元素 $a_{kk}^{(k)} = 0$ 的情况，但是方程组(1)有唯一解，则在 $a_{k+1,k}^{(k)}$，$a_{k+2,k}^{(k)}$，$\cdots$，$a_{nk}^{(k)}$ 中至少有一个不为零，比如 $a_{k+j,k}^{(k)} \neq 0$，那么将第 k 个方程与第 $k+j$ 个方程对调，便可继续进行消去。有时虽然 $a_{kk}^{(k)} \neq 0$，但其绝对值很小，以它作除数也会引起较大误差，从而使最后的解不精确，甚至面目全非。为了避免这种情况，应将 $|a_{k+1,k}^{(k)}|$，$|a_{k+2,k}^{(k)}|$，$\cdots$，$|a_{nk}^{(k)}|$ 中最大者所在方程与第 k 个方程对调。这样的消去过程称为列主元消去法。

三角分解法 该方法是直接从方程组(2)出发，将系数矩阵 $\boldsymbol{A}$ 分解为两个三角形矩阵之积，即 $\boldsymbol{A} = \boldsymbol{LU}$。当 $\boldsymbol{L}$ 为单位下三角形矩阵，$\boldsymbol{U}$ 为上三角形矩阵时，这种分解称为杜利特尔分解法；当 $\boldsymbol{L}$ 为下三角形矩阵，$\boldsymbol{U}$ 为单位上三角形矩阵时，称为克劳特分解法；如果 $\boldsymbol{A}$ 为对称正定矩阵，则 $\boldsymbol{U} = \boldsymbol{L}^{\mathrm{T}}$，相应的分解法称为楚列斯基分解法或对称分解法。以杜利特尔和楚列斯基分解法为例说明。在杜利特尔分解中，矩阵 $\boldsymbol{L}$ 和 $\boldsymbol{U}$ 的元素 l_{ij} 和 u_{ij} 由下列递推公式计算。对 $k = 1, 2, \cdots, n$,

$$\left.\begin{aligned} u_{kj} &= a_{kj} - \sum_{m=1}^{k-1} l_{km}u_{mj}, \quad j = k, k+1, \cdots, n \\ l_{ik} &= \Big(a_{ik} - \sum_{m=1}^{k-1} l_{im}u_{mk}\Big)\Big/u_{kk}, \quad i = k+1, k+2, \cdots, n \end{aligned}\right\} \tag{10}$$

其中令 $\sum_{m=1}^{0} = 0$。公式(10)的计算顺序是：先计算 $\boldsymbol{U}$ 的第1行，再计算 $\boldsymbol{L}$ 的第1列，然后计算 $\boldsymbol{U}$ 的第2行，再计算 $\boldsymbol{L}$ 的第2列，如此继续下去，直到得到 $\boldsymbol{L}$ 和 $\boldsymbol{U}$ 为止。此时，式(2)可写为两个方程组

$$\boldsymbol{Ly} = \boldsymbol{b}, \quad \boldsymbol{Ux} = \boldsymbol{y} \tag{11}$$

再分别解这两个方程组，得到式(2)的近似解。

为避免除数 u_{kk} 为零或其绝对值过小而带来误差，同样可在得到 $\boldsymbol{U}$ 的元素 u_{kk}，$u_{k,k+1}$，$\cdots$，u_{kn} 后，在它们中选主元，然后进行对调，而 x 的分量也进行相应对调。

在楚列斯基分解中，下三角形矩阵 $\boldsymbol{L}$ 及其转置矩阵 $\boldsymbol{L}^{\mathrm{T}}$ 的元素 l_{ik} 由下列公式计算。对 $k = 1, 2, \cdots, n$,

$$\left.\begin{aligned} l_{kk} &= \Big(a_{kk} - \sum_{m=1}^{k-1} l_{km}^2\Big)^{1/2}, \\ l_{ik} &= \Big(a_{ik} - \sum_{m=1}^{k-1} l_{im}l_{km}\Big)\Big/l_{kk}, \\ i &= k+1, k+2, \cdots, n \end{aligned}\right\} \tag{12}$$

由于 $\boldsymbol{A}$ 对称正定，l_{kk} 均不为零，故不必选主元。但在式(12)中有开方运算，工作量大，可采用改进的楚列斯基分解法。

迭代法

考虑线性代数方程组

$$\boldsymbol{Ax} = \boldsymbol{b} \tag{13}$$

其中 $\boldsymbol{A}$ 为非奇异矩阵。迭代法的基本思想是：从某个初始向量$\boldsymbol{x}^{(0)}$出发，构造一个向量序列$\{\boldsymbol{x}^{(k)}\}$，使其收敛于某个极限向量$\boldsymbol{x}^*$，且$\boldsymbol{x}^*$就是方程组(13)的准确解。

迭代法分点迭代法和块迭代法两种。点迭代法是指每次从已有的近似解分量求出一个新的近似解分量，如此逐个地求下去，直到求出全部分量为止。然后，重复此过程。块迭代法则是先将系数矩阵 $\boldsymbol{A}$，解向量 $\boldsymbol{x}$ 和右端项 $\boldsymbol{b}$ 进行分块，然后将每一子块视为一个元素，并按点迭代法的类似公式进行迭代。下面介绍几种点迭代法。

雅克比迭代法 对于一般的线性方程组(13)，设 $a_{ii}(i = 1, 2, \cdots, n)$ 均不为零，式(13)的雅克比迭代格式为

$$x_i^{(k+1)} = \frac{b_i}{a_{ii}} - \sum_{\substack{j=1 \\ j \neq i}}^{n} \frac{a_{ij}}{a_{ii}} x_j^{(k)}, \quad i = 1, 2, \cdots, n; k = 1, 2, \cdots \tag{14}$$

若将系数矩阵 $\boldsymbol{A}$ 分裂为下列形式

$$\boldsymbol{A} = \boldsymbol{D} + \boldsymbol{L} + \boldsymbol{U},$$

其中 $\boldsymbol{D}$ 为对角矩阵，$\boldsymbol{L}$ 和 $\boldsymbol{U}$ 分别为对角元素为零的下三角和上三角矩阵。用 $\boldsymbol{D}+\boldsymbol{L}+\boldsymbol{U}$ 代替矩阵 $\boldsymbol{A}$，则式(14)的矩阵向量形式为

$$x^{(k+1)} = \boldsymbol{D}^{-1}\boldsymbol{b} - \boldsymbol{D}^{-1}(\boldsymbol{L}+\boldsymbol{U})\boldsymbol{x}^{(k)},\quad k = 1,2,\cdots \tag{15}$$

高斯-赛德尔迭代法　在迭代格式(14)中，将第 i 个方程迭代解出的 $x_i^{(k+1)}$ 值代替其后各方程中的 $x_i^{(k)}$，从而得到高斯-赛德尔迭代格式

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \times \Big(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}\Big),$$

$$i = 1,2,\cdots,n;k = 1,2,\cdots \tag{16}$$

式(16)的矩阵向量形式为

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{D}^{-1}\boldsymbol{b} - \boldsymbol{D}^{-1}(\boldsymbol{L}\boldsymbol{x}^{(k+1)} + \boldsymbol{U}\boldsymbol{x}^{(k)}),\quad k = 1,2,\cdots \tag{17}$$

松弛法　该方法分同时松弛法和逐步超松弛法两种。它是利用矩阵正定性来加快迭代过程的收敛速度。同时松弛法(也称 JOR 方法)迭代格式为

$$x_i^{(k+1)} = x_i^{(k)} + \frac{\omega}{a_{ii}}\Big(b_i - \sum_{j=1}^{n} a_{ij}x_j^{(k)}\Big) \tag{18}$$

令 $\boldsymbol{B}=\boldsymbol{D}-\boldsymbol{A}$，则式(18)的矩阵向量形式为

$$\boldsymbol{x}^{(k+1)} = (1-\omega)\boldsymbol{x}^{(k)} + \omega\boldsymbol{D}^{-1}\boldsymbol{B}\boldsymbol{x}^{(k)} + \omega\boldsymbol{D}^{-1}\boldsymbol{b} \tag{19}$$

其中 ω 为松弛因子。当 $\omega>1(\omega<1)$ 时称为超松弛(低松弛)。

逐次超松弛法(也称 SOR 法)的迭代格式为

$$x_i^{(k+1)} = x_i^{(k)} + \frac{\omega}{a_{ii}}\Big(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i}^{n} a_{ij}x_j^{(k)}\Big) \tag{20}$$

此式的矩阵向量形式为

$$(\boldsymbol{D}+\omega\boldsymbol{L})\boldsymbol{x}^{(k+1)} = (1-\omega)\boldsymbol{D}\boldsymbol{x}^{(k)} - \omega\boldsymbol{U}\boldsymbol{x}^{(k)} + \omega\boldsymbol{b} \tag{21}$$

式(19)和式(21)可统一写为

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{G}\boldsymbol{x}^{(k)} + \boldsymbol{f} \tag{22}$$

其中 $\boldsymbol{G}$ 称为迭代矩阵。对应于式(19)和式(21)分别有

$$\boldsymbol{G} = \boldsymbol{D}^{-1}[(1-\omega)\boldsymbol{D} + \omega\boldsymbol{B}]$$

和

$$\boldsymbol{G} = (\boldsymbol{D}+\omega\boldsymbol{L})^{-1}[(1-\omega)\boldsymbol{D} + \omega\boldsymbol{U}]$$

$\boldsymbol{f}$ 分别为 $\omega\boldsymbol{D}^{-1}b$ 和 $(\boldsymbol{D}+\omega\boldsymbol{L})^{-1}\omega b$。

松弛因子 ω 的选取是很重要的。实际计算中，往往是在 ω 所在范围内选取若干值试迭代，从中选取使收敛速度快的 ω 值。

下面给出三个迭代法收敛定理。

定理 1　若方程组(13)的系数矩阵 $\boldsymbol{A}$ 为严格对角占优矩阵，则 $\boldsymbol{A}$ 为非奇异矩阵，且对任意初值 $\boldsymbol{x}^{(0)}$，方程组(13)的雅克比迭代法和高斯-赛德尔迭代法收敛。

定理 2　松弛法迭代格式(22)收敛的充要条件为 $\rho(\boldsymbol{G})<1$，其中 $\rho(\boldsymbol{G})$ 为矩阵 $\boldsymbol{G}$ 的谱半径。

定理 3　若 $\boldsymbol{A}$ 为对称正定矩阵，且 $0<\omega<2$，则逐次松弛法收敛。

除上述方法外，解线性代数方程组的方法还有克雷洛夫子空间方法(例如共轭梯度法)、蒙特卡罗法、解三对角方程组的追赶法等。对于一些特殊线性方程组，如由偏微分方程离散化得到的线性方程组，可用对称逐次超松弛法和隐式交替方向法等迭代方法求解。

关于线性方程组的求解，目前国际上提供了著名软件库 LINPACK 可供用户使用。

参考文献

1. 冯康，等. 数值计算方法. 北京：国防工业出版社，1978.

2. Wilkinson J H, Reinsch C. Handbook of Automatic Computation Linear Algebra 2. Berlin: Spring Verlag, 1971.

3. Golub G H, van Loan C F. 矩阵计算. 3 版. 袁亚湘，等译. 北京：人民邮电出版社，2011

（李晓梅　安恒斌）

xianxing luoji

线性逻辑（linear logic）　由法国数理逻辑学家、计算机科学家 J. Y. Girard 于 1987 年建立的一种非经典的逻辑系统。它源于 G. Gentzen 提出的矢列演算，其特点是推理规则作用于公式序列上，而不仅是单个公式上。此外，它将推理规则明确区分为逻辑推理规则及结构性推理规则。在 4 条结构性规则中，切割规则：$\dfrac{\underline{A}\vdash C,\underline{B}\quad \underline{A}',C\vdash \underline{B}'}{\underline{A},\underline{A}'\vdash \underline{B},\underline{B}'}$，其中 $\underline{A}$，$\underline{A}'$ 等表示公式的序列，在证明中可以消去，这就是著名的甘岑定理。减弱规则，例如 $\dfrac{\underline{A}\vdash \underline{B}}{\underline{A}\vdash \underline{B},\underline{C}}$；以及收缩规则，例如 $\dfrac{C,C,\underline{A}\vdash \underline{B}}{C,\underline{A}\vdash \underline{B}}$ 的使用，意味着证明的前提在证明中可使用无穷多次。因而经典逻辑是不可构造的。另一条交换规则：$\dfrac{\underline{A}\vdash \underline{B},C,D,\underline{B}'}{\underline{A}\vdash \underline{B},D,C,\underline{B}'}$ 对于系统的构

造性是无害的。线性逻辑与经典逻辑的区别在于前者去除了破坏构造性的两条结构性规则。但其他逻辑联结词的规则需要调整和修正。例如在经典逻辑中，以下两条规则 $\frac{\underline{A}\vdash C,\underline{B}\quad \underline{A}'\vdash D,\underline{B}}{\underline{A},\underline{A}'\vdash C\wedge D,\underline{B}}$ 以及 $\frac{\underline{A}\vdash C,\underline{B}\quad \underline{A}'\vdash D,\underline{B}'}{\underline{A},\underline{A}'\vdash C\wedge D,\underline{B},\underline{B}'}$，在系统中包含所有结构性规则的情况下是等价的。而在线性逻辑中就需要引进不同的合取联结词：乘合取⊗及加合取 & 来表述上面的两条规则。对偶地，需要引进乘析取 以及加析取⊕。关于线性否定⊥的定义如下：

每个原子公式都有两种形式 A 及 $A^{\perp}$。A 的否定为 $A^{\perp}$，$A^{\perp}$ 的否定为 A，即 $(A^{\perp})^{\perp}=A$。

用德·摩根定律定义带联结词的复合公式的线性否定：

$(A\otimes B)^{\perp}=A^{\perp}\quad B^{\perp}$，　$(A\&B)^{\perp}=A^{\perp}\oplus B^{\perp}$

$(A\quad B)^{\perp}=A^{\perp}\otimes B^{\perp}$，　$(A\oplus B)^{\perp}=A^{\perp}\&B^{\perp}$

线性否定本身并不是联结词，例如，$(A\otimes B^{\perp})^{\perp}$ 仅仅是 $A^{\perp}\quad B$ 的另一种表示，后者还可记为线性蕴含式 $A\quad B$。

为简单起见，线性逻辑的序列演算使用单边序列 $\vdash A_1^{\perp},\cdots,A_n^{\perp},B_1,\cdots,B_m$ 代替甘岑系统中的双边序列 $A_1,\cdots,A_n\vdash B_1,\cdots,B_m$。

线性逻辑系统的常项符号有⊥，1，⊤，0。

公理及推理规则可表述如下：恒同公理 $\vdash A^{\perp},A$；加公理 $\vdash T,\underline{A}$；乘公理 $\vdash 1$。切割规则 $\frac{\vdash C,\underline{A}\quad \vdash C^{\perp},\underline{B}}{\vdash \underline{A},\underline{B}}(\text{Cut})$；交换规则 $\frac{\vdash \underline{A},C,D,\underline{B}}{\vdash \underline{A},D,C,\underline{B}}(\times)$。

逻辑规则表述如下：关于加的合取及析取

$$\frac{\vdash A,\underline{C}\quad \vdash B,\underline{C}}{\vdash A\&B,\underline{C}}(\&);\quad \frac{\vdash C,\underline{A}}{\vdash C\oplus D,\underline{A}}(1\oplus);$$

$$\frac{\vdash D,\underline{A}}{\vdash C\oplus D,\underline{A}}(2\oplus)$$

关于乘的合取及析取

$$\frac{\vdash C,\underline{A}\quad \vdash D,\underline{B}}{\vdash C\otimes D,\underline{A},\underline{B}}(\otimes);\quad \frac{\vdash C,D,\underline{A}}{\vdash C\quad D,\underline{A}}(\quad);$$

$$\frac{\vdash \underline{A}}{\vdash \perp,\underline{A}}(\perp)$$

也可加入对偶的模态联结词！A(当然 A)和？A(为何不是 A)扩充线性逻辑系统。其否定为 $(!\ A)^{\perp}=?\ A^{\perp}$，$(?\ A)^{\perp}=!\ A^{\perp}$。

关于模态词的推理规则为

$$\frac{\vdash B,?\ \underline{A}}{\vdash !\ B,?\ \underline{A}}(!);\quad \frac{\vdash \underline{A}}{\vdash ?\ B,\underline{A}}(W?)$$

$$\frac{\vdash ?\ B,?\ B,\underline{A}}{\vdash ?\ B,\underline{A}}(C?);\quad \frac{\vdash B,\underline{A}}{\vdash ?\ B,\underline{A}}(D?)$$

系统中同样可引进谓词，但线性逻辑的主要特征已表现在命题演算部分中。

线性逻辑具有两种含义完全不同的语义。Tarski 传统的语义只注重逻辑演算的结果，而与证明过程无关。线性逻辑的 Tarski 传统语义称为相空间语义，由可换幺半群 $(\boldsymbol{P},\cdot,1)$ 以及一个取定的 $\boldsymbol{P}$ 的子集⊥构成，记为 $(\boldsymbol{P},\cdot,1,\perp)$，将线性逻辑中的常项、公式解释为相空间中的特定子集，将逻辑符号解释为相空间中的运算。如果一个公式 A 在相空间 $\boldsymbol{P}$ 中的解释为 $\tilde{A}$，且 $1\in\tilde{A}$，就称 A 在 $\boldsymbol{P}$ 中有效。线性逻辑演算(不包含！及？)相对于其在相空间中的有效性是可靠且完全的。另一种更重要的语义是由 Girard 重新开拓的 Heyting 传统的语义。其目的是建立证明的模型。例如假定原子公式的证明已知，公式 $A\wedge B$ 的证明为由 A 的证明 p 及 B 的证明 q 构成的序对 (p,q)；$A\to B$ 的证明是一个函数 f，它将 A 的每一个证明 p 映射到 B 的证明 $f(p)$。线性逻辑的紧连空间语义就实现了 Heyting 的上述思想。首先将每个命题及常项对应到紧连空间，由联结词构造出的每个公式也归纳地对应到相应的紧连空间。例如 X,Y 分别为公式 A,B 对应的紧连空间，则 $X\quad Y$ 定义为由 X 至 Y 的全体线性函数的轨迹构成的紧连空间，此紧连空间对应公式 $A\quad B$。我们用相同的符号表示 A 式及它所对应的紧连空间。在这种语义下可以证明：若 π 是线性逻辑中 $\vdash A$ 的一个证明，则存在紧连空间 A 中唯一的对象 π^*，π^* 描述了证明 π。此定理说明，一个公式的可证性是由公式的内在结构完全决定的。

线性逻辑的另一重要结果是建立了证明的范式，称为证明网络。证明网络可看作是线性逻辑的自然推理系统。由于线性蕴含 $A\quad B=A^{\perp}\quad B$，证明网络中就可避免使用经典逻辑中的蕴含引入和消去规则，具有十分规范的形式。

线性逻辑对于构造性问题的成功研究为证明论及计算机科学理论中提出的一些基本问题开辟了新的途径。

参考文献

1. Girard J Y. Linear logic. Theoretical Computer Science, 1987, (50)

2. Girard J Y, Lafont Y, Taylor P. Proofs and

types. Cambridge University Press, 1989　（黄且圆）

xianxing wenfa

线性文法（linear grammar）　一种受限的**上下文无关文法**。如果上下文无关文法 $G=(\Sigma,V,S,P)$，P 中的所有生成式为形式 $A\to u$ 或者 $A\to uBv$；$A,B\in V$；$u,v\in\Sigma^*$，则称 G 为线性文法。线性文法生成的语言称为线性语言。例如 $G_1=(\{a,b\},\{S\},S,P_1)$，其中 $P_1=\{S\to aSb, S\to ab\}$，是线性文法，生成的线性语言 $L(G_1)=\{a^nb^n\mid n\geqslant 1\}$。正则语言类是线性语言类的真子类，而线性语言类是上下文无关语言类的真子类。

超线性文法　线性文法的一种扩展。设上下文无关文法 $G=(\Sigma,V,S,P)$，如果存在有限多个两两无交的（可能为空的）变量集合 $V_0,V_1,\cdots,V_n$，使得 $V=\bigcup_{i=0}^{n}V_i$ 且对于每个 V_i 及任意的 $A\in V_i$，P 中的所有生成式为形式 $A\to\alpha$，$\alpha\in(\Sigma\cup V_0\cup V_1\cup\cdots\cup V_{i-1})^*$，或者 $A\to uBv$，$B\in V_i$ 且 $u,v\in\Sigma^*$，则称 G 为超线性文法。特别地，$n=0$ 时，G 为线性文法。超线性文法生成的语言称为超线性语言。例如 $G_2=(\{a,b\},V,S,P_2)$，其中 $V=\{S,A,B\}$，$P_2=\{S\to AB, A\to aAb, A\to ab, B\to aBb, B\to ab\}$。取 $V_1=\{S\}$，$V_0=\{A,B\}$，则 $V=V_0\cup V_1$，且 P_2 中的生成式均为超线性文法要求的形式，所以 G_2 是超线性文法，生成的语言 $L(G_2)=\{a^nb^n,a^mb^m\mid n\geqslant 1,m\geqslant 1\}$ 是超线性语言。

有限转向的下推自动机　S. Ginsburg 等人于 1965 年提出的**下推自动机**的一种限制类型。以空存储接受的下推自动机 $M=(Q,\Sigma,\Gamma,\delta,q_0,Z_0,\varnothing)$ 在接受一个字的过程中，栈符号由 Z_0 开始，栈符号个数时增时减，最后排空。栈符号个数由增加到减少或者由减少到增加均称为一次转向。如果下推自动机 M 接受的每一个字转向次数均不超过 $2k-1$，k 为某正整数，则称 M 是 $2k-1$ 次转向的下推自动机。有限转向的下推自动机 M 接受的语言用 Null(M) 表示。例如 $M=(\{q\},\{a,b\},\{Z_0,A,B,a,b\},\delta,q,Z_0,\varnothing)$，其中 $\delta(q,a,a)=\{(q,\lambda)\}$，$\delta(q,b,b)=\{(q,\lambda)\}$，$\delta(q,\lambda,Z_0)=\{(q,AB)\}$，$\delta(q,a,A)=\{(q,Ab),(q,b)\}$，$\delta(q,a,B)=\{(q,Bb),(q,b)\}$，是一个 3 次转向的下推自动机，它接受的语言 Null$(M)=\{a^nb^n,a^mb^m\mid n\geqslant 1,m\geqslant 1\}$。对于任意的有限转向的下推自动机 M，可以构造一个超线性文法 G，使得 $L(G)=\text{Null}(M)$。反之亦然。从而有限转向的下推自动机接受的语言类和超线性语言类是同一语言类。特别地，1 次转向的下推自动机接受的语言类等同于线性语言类。

参考文献

Harrison M A. Introduction to formal language theory. Boston, MA: Addison - Wesley, 1978

（郭清泉）

xianxing youjie zidongji

线性有界自动机（linear bounded automaton）　输入带上以特殊符号 ⊄、$ 分别作左、右端标识，读头只在 ⊄ 和 $ 之间移动，且在 ⊄、$ 处不能打印任何其他符号的非确定**图灵机**。

$M=(Q,\Sigma,\Gamma,\delta,q_0,⊄,\$,F)$ 称为非确定线性有界自动机（简记为 LBA），其中 Q 为状态的有限集合，Γ 为带上允许符号的有限集合，$\Sigma\subseteq\Gamma$，Σ 为输入符号集，$\delta:Q\times\Gamma\to Q\times\Gamma\times\{L,R\}$ 为转移函数，L 表示左移一格，R 表示右移一格，$q_0\in Q$，q_0 称为初始状态，⊄，$ $\in\Sigma$，F 为终止状态集，$F\subseteq Q$。M 所接受的语言 $L(M)=\{\underline{x}\mid\underline{x}\in(\Sigma-\{⊄,\$\})^*$，且 $q_0⊄\underline{x}\$\overset{*}{\underset{M}{\vdash}}\alpha q\beta$，$q\in F$，$\alpha,\beta\in\Sigma^*\}$（$\overset{*}{\underset{M}{\vdash}}$ 意指：从 q_0 经有限次转移函数作用于字 $\underline{x}$ 可到达 q），称为上下文有关语言或 1 型语言，记为 CSL。若上述定义中采用确定型图灵机，则称 M 为确定型线性有界自动机（DLBA），并称 $L(M)$ 为确定的上下文有关语言（DCSL）。CSL 类是递归可枚举（0 型）语言类的真子集，上下文无关（2 型）语言类是 CSL 类的真子集。

上下文有关文法　$G=(V,\Sigma,S,P)$，V 为变元的有限集，Σ 为终结符的有限集，$V\cap\Sigma=\varnothing$，S 为开始符号，$S\in V$，P 为产生式的有限集，P 中产生式呈 $\alpha\to\beta$ 形且 $|\alpha|\leqslant|\beta|$，$|\alpha|\neq 0$（即 α 不是空字），其中 $\alpha,\beta\in(V\cup\Sigma)^*$，$|\alpha|$ 表示字 α 所含符号的个数，$G=(V,\Sigma,P,S)$ 称为上下文有关文法（CSG），G 生成的语言 $L(G)$ 必为某 LBA 所接受（识别）。例如 $G=(\{S,A,B,C,D,E,H,K\},\Sigma=\{\underline{a}.\ eps,+,-,*,/,(,)\}$，$S$ 为开始符号，$P=\{S\to A, A\to B)\mid\underline{a},B\to(C\mid BC\mid(A,C)\to D),CH\to DH,D\to HA\mid EA,E\to DK,H\to+\mid-,k\to*\mid/\}$，$G$ 是一个 CSG，$L(G)$ 生成的是操作数为 $\underline{a}$，以中缀算子优先的算术表达式。CSG 的产生式可转化为 Kurada 形：$A\to\underline{a}\mid B\mid BC,AB\to CD$，其中 $A,B,C,D\in V$，$\underline{a}\in\Sigma$；或转化为右（左）CSG，其产生式为 $A\to\underline{a}\mid B\mid BC,AB\to AC(AB\to CB)$

形。这些文法是彼此等价的。右(左)CSG 可用作 CSL 的字处理工具。特别当 $\Sigma=\{0,1\}$ 时,CSG 为"标准化"的,标准 CSG 的集合是可数的。CSL 集对并、交、连接、正闭包、替换、逆同态、CYCLE、Reversal 等运算封闭,但对补、MIN、MAX 还不知道是否封闭(封闭是指语言运算应用于 CSL 后结果仍为 CSL)。给完字 $\underline{x}$,CSL L_1,L_2,正规语言 R(参见**正规文法**),$\underline{x}\in L_1$ 是可以判定的,但 $L_1=\varnothing$? $L=\Sigma^*$? $L_1=L_2$? $L_1\subseteq L_2$? $L_1\cap L_2=\varnothing$? $L_1=R$? 等都是不可判定的。

LBA 与 DLBA 接受的语言集是否相同是目前尚未解决的著名问题,称为 LBA 问题。

参考文献

Hopcroft J E, Ullman J D. Introduction to automata theory, languages, and computation. Reading, MA: Addison - Wesley, 1979 (张一立)

xiangbian suiji cunchuqi

相变随机存储器(phase-transition random access memory) 参见**非易失新型半导体存储器**。

xiangwei tiaozhi jishu

相位调制技术(phase modulation technology) 相位调制技术较少用于模拟信号源,主要用于数字信号源的数字调相。

数字调相,又称相移键控法(phase-shift keying, PSK),指高频载波信号的相位随调制信号的二进制数值变化而变化,但载波信号的振幅不变的调制技术。按照相位状态数量的不同,可以将相位调制技术分为二相调制技术(2-PSK)、四相调制技术(4-PSk)和十六相调制技术(16-PSK)等。

二相调制的过程举例说明如下。设调制载波用公式 $s(t)=a(t)\cos\{2\pi(f_c+f_m)t+\phi_m(t)\}$ 表示。当载波传输信号时,使用三个调制参数:振幅 $a(t)$、调制频率 f_m、相位 $\phi_m(t)$。在二相调制中,假设振幅 $a(t)$ 为 A,调制频率 f_m 为 0,则二相调制 2-PSK 传输信号可由下式表示

$$s(t)=A\cos(2\pi f_c t+a_i\pi)\begin{cases}a_i=0\\a_i=1\end{cases}$$

图 1 是二相调制的图示。

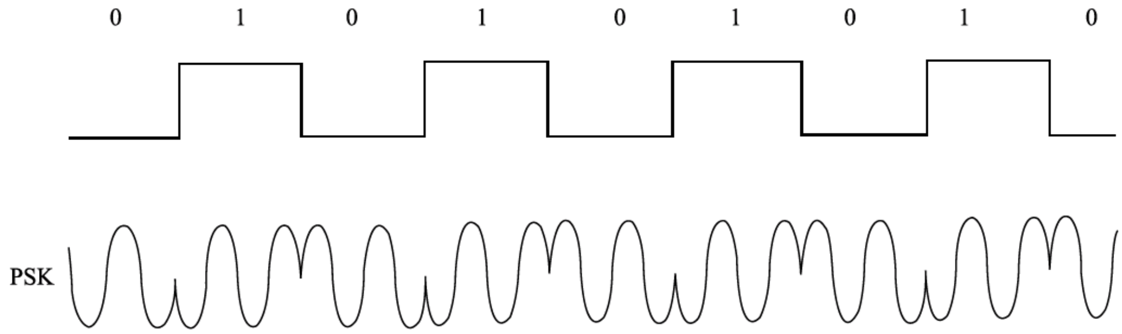


图 1 二相 PSK

在现代数字通信传输系统中,相位调制技术通常与幅度调制技术一起使用,称为正交幅度调制(quadrature amplitude modulation, QAM)。

参考文献

1. [日]关清三. 数字调制解调基础. 北京: 科学出版社, 2002

2. 蒋青, 于秀兰. 通信原理. 北京: 人民邮电出版社, 2008 (董守斌 张凌)

xiangliangchang keshihua

向量场可视化(vector field visualization) 将向量场蕴含的各种现象和结果以可视、直观的图形方式展现出来以揭示向量场运动变化规律的可视化方法。

向量场是科学计算和工程应用中一种常见的数据类型。与标量场相比,向量场数据不仅有大小,而且具有方向,其高维、拓扑复杂、动态性等特点,使得大规模三维向量场的可视化极具挑战性。现有的向量场可视化方法一般被分为以下三类:

(1) 基于几何元素的方法 这类方法大多从实验型流场可视化技术演变而来,是实验方法的一种计算机仿真。常用的方法包括:①点图标方法:最常用的是箭头,通常取箭头的大小/长度与向量的大小成正比,方向指向向量的方向,通过透视变换和消隐可在一定程度上反映三维的信息。该方法简单、易于实现,但难以揭示出数据的内在连续性。②向

量线/面方法：如场线、迹线、脉线、流线、粒子、向量面、向量管等，目前生成方法主要分为基于数值积分以及基于流函数构造两大类。向量线/面方法直观、生成速度快，但由于建立在离散采样的基础上，当采样密集时，会造成视觉上的混乱，而采样稀疏时，有时又会漏掉向量场中重要的特征和细节，很难全面、连续地揭示大规模向量场的特性。

（2）基于纹理的方法　这类方法通过将方向的信息映射到随机的纹理上来表达矢量方向，如点噪声（spot noise）、线积分卷积（line integral convolution）法等，由于纹理映射在图像空间进行，因而具有图像空间的连续性，克服了基于几何元素映射方法的采样问题，即使在向量方向变化很大的区域也能较好地揭示向量的方向。该类方法虽然在二维以及三维曲面上取得了好的绘制效果，却始终难以成功地拓展到三维，三维纹理之间众多的信息叠加到一起互相遮挡可造成严重的视觉混乱现象。

（3）基于特征的方法　目前这方面的研究可大致分为以下三类：①流场拓扑分析法：通过提取向量场的临界点并根据临界点附近向量场的特性将临界点分为交点、聚点、马鞍点及中心点等，从而建立向量场的拓扑结构；②向量场典型特征结构的抽取、跟踪，包括对涡核、激波的抽取等；③向量场的聚类和简化方法：通过聚类和简化，突出向量场的特征，提高可视化的质量。虽然这类方法一直是近些年可视化领域研究的热点，但由于特征的定义以及抽取特征的方法均与具体的应用领域密切相关，尚未找到通用有效的方法。

综上所述，探索更有效的既可最大限度揭示向量场的信息又能避免可视混乱现象的向量场可视化方法依然是目前需要研究的课题，其中涉及探索蕴含更丰富信息的向量场可视绘制模型、减少绘制数据量的大规模向量场特征可视化方法以及基于图形硬件的实时大规模向量数据高质量绘制方法等。

参考文献

1. Hansen C D, Johnson C R. The visualization handbook. Elsevier, 2005

2. 唐荣锡，汪嘉业，彭群生，等. 计算机图形学教程（修订版）. 北京：科学出版社，2000　（陈莉）

xiangliang chuli bujian

向量处理部件（vector processing unit）　计算机中专用于执行向量计算的功能单元。基本的向量运算包括：对一个向量的各分量执行同一运算；对同样维数的两个向量的对应分量执行同一运算；一个向量的各分量都与同一标量执行同一运算；等等，均可产生一个新的向量。此外，可在一个向量的各分量间执行某种运算，如连加、连乘或连续比较等操作，使之产生成一个标量。

20 世纪 70 年代，出现了专用于科学与工程计算的**阵列处理机**和向量处理机。二者都属于单指令流多数据流（SIMD）计算机，其中阵列处理机用一组同步工作阵列以空间重复方式进行多个向量分量的并行运算，向量处理机用深度流水线结构以时间重叠方式进行多个向量分量的并发运算。到了 80 年代，有一些大型计算机开始配备可选的流水线向量处理部件，增强其在科学计算领域的竞争力，如 IBM 3090 的 VF（vector facility）部件。

早期的**微处理器**因为性能有限，较少用于科学工程计算，故也没有配备向量处理部件。20 世纪 90 年代以后，为了提高通用微处理器处理图像、音频、视频等多媒体信息的效率，出现了专用的多媒体扩展部件（参见**多媒体扩展部件**）。多媒体扩展部件虽然在形式上具备了向量处理部件的基本要素（如专用的扩展指令集、以 SIMD 方式执行针对多个分量数据的同时运算等），但因其基本数据分量只是短字长（字节、半字等）的定点数据，并不能胜任传统向量处理所面向的科学工程计算任务，所以也很少有人把多媒体扩展部件看作向量处理部件。

2000 年前后，Intel 公司把 64 位的 MMX 多媒体扩展部件扩展为 128 位的 SSE，支持 4 个单精度浮点运算的同时执行，表明通用微处理器内部的向量处理部件已经初见端倪。随后，SSE 2/3/4 相继推出并被 AMD 公司采纳，证明了向量处理部件对提高微处理器性能的重要作用。2011 年 Intel 正式把 256 位宽的 SIMD 指令集及运算部件命名为“先进向量扩展”（advanced vector extensions，AVX）。

向量处理的关键问题是存储器系统能否满足运算部件的带宽需求。根据传统向量处理机的经验，寄存器-寄存器结构（源操作数来自、结果存入向量寄存器）比存储器-存储器结构（源操作数来自、结果存入主存储器）具有更高的效率，因此，现代微处理器中的向量处理部件都采用了寄存器-寄存器结构，并希望充分利用片上高速缓存带来的低延迟、高带宽特性。

传统的流水线向量处理器偏爱长向量，以弥补深度流水线的启动开销。例如，Cray 1 的向量寄存器长度为 64 分量。由于现代微处理器受到功耗的

限制,已难以继续利用深度流水线带来的高主频来提高向量处理性能,故这类处理器中的向量处理部件都转向类似于阵列处理器的结构,即用多个普通的(流水)运算器的同时执行来提高性能。考虑到运算器的数量和数据通路的宽度,向量长度(或向量寄存器的宽度)相对短一些。例如,AVX 的宽度为 256 位,相当于 4 个双精度分量或 8 个单精度分量。

今后,微处理器中向量处理部件的位宽还会逐步增大,例如,Intel MIC 的向量数据通路已达 512 位。此外,一些图形处理器中也实现了 128/256 位宽的向量指令。

参考文献

Hennessy J L, Patterson D A. 计算机体系结构——量化研究方法. 5 版. 北京:机械工业出版社,2012

(张悠慧　唐志敏)

xiangliang jisuan

向量计算(vector computing)　以向量为单位进行的计算。相对于标量而言,向量是指一组同类型标量的有序集合,其中每个标量称为该向量的一个分量或元素。向量计算与标量计算的主要区别在于,一条标量指令仅处理一个或一对操作数,一条向量指令则可以处理 $n(n \geqslant z)$ 个或 n 对操作数,因此向量计算提供了更大的并行性。

执行向量计算的计算机称为向量计算机。向量计算机通常由一个标量处理机附加一个向量处理机组成。标量处理机负责所有指令的加载、译码和标量指令的执行,向量处理机负责向量指令的执行。按操作数的来源,向量计算机分为存储器-存储器型和寄存器-寄存器型。向量计算机对于有规则的大规模数值计算,大大提高了运算速度,主要依赖于下列技术的实现:采用流水线结构,促成运算的连续性;设置多个功能部件,并行处理多条向量指令;运用"链接"功能,使一个流水线部件的结果直接作为另一个流水线部件的操作数,不存在中间结果的存取问题;合理的存储器层次结构或通过流水存取方式提高存储器带宽,与运算部件的速度相匹配;减少程序中辅助指令条数,提高程序整体运行效率。

第一个向量计算机 CDCStar-100 出现于 1973 年。1976 年问世的 Cray-1 向量计算机被公认为世界上第一个超级计算机,每秒可产生一亿个浮点结果。此后,向量计算机成为高性能计算的主流产品,一直维持到 20 世纪 90 年代初期。其间,Cray 公司先后研制出 CrayXMP,CrayYMP,CrayC90 和 CrayT94 等。向量处理在其发展过程中,逐步与**大规模并行处理**相结合,出现了并行向量处理(PVP)体系结构,进一步提高向量计算机整体性能。PVP 体系结构通常以高性能**互连网络**连接多个结点,每个结点可包含一个或多个高性能向量 CPU。NEC 公司的 SX 系列、富士通公司的 VPP 系列、Cray 公司的 SV1 和 SV2 都采用 PVP 结构。SX-5 和 VPP5000 均可达 512 个 CPU,峰值速度分别为 4 T FLOPS 和 4.9 T FLOPS。SV1 扩展型最多可达 1024 个 CPU, 峰值速度为 2 T FLOPS。2002 年 3 月日本 NEC 等单位宣布研制完成的"地球模拟器"是 21 世纪初期最快的并行向量超级计算机。该系统自行开发的向量 CPU 达到 8 G FLOPS,每个结点 8 个 CPU,全系统 640 个结点共 5120 个 CPU,峰值速度 40 T FLOPS,存储器总容量 10 TB,互连网络为单层交叉开关。

另一个发展是标量处理器中加入了伪向量处理技术。所谓**伪向量处理**,是在足够存储器带宽的前提下,将数据从主存储器直接加载到浮点寄存器,避开因高速缓存容量问题引起供数不足的矛盾,借助滑动窗口技术实现每个时钟周期产生一个浮点结果的目的。日立公司的 SR2201 和 SR8000 系统采用了这项技术。

总体来看,向量计算机因其定制 CPU 和定制互连网络等因素,性能价格比远不敌商品化的高性能通用微处理器和互连产品构成的集群计算系统,自 20 世纪 90 年代以来一直在萎缩,这种趋势还将继续下去。唯有日本仍对向量机的发展给予相当重视,这是由于日本在传统上研究向量机的力量较强,以及日本研制向量处理器的微电子技术力量较强。

参考文献

1. 郑纬民,汤志忠. 计算机系统结构. 2 版. 北京:清华大学出版社,1998

2. Kai Hwang. Advanced Computer Architecture. McGraw-Hill-Inc., 1993

(桂亚东　谢向辉)

xiaoxi chuandi

消息传递(message passing)　在分布式并行计算机系统中,通过传递消息包来实现其中各计算机之间的通信和同步的一种机制。

一个消息可以是一个很短的同步信息,也可能是一个长度为数兆字节的数据文件。在通常情况下,消息是由一组消息包组成的。消息包是 1 次发送或

接收的单位。它由包头、包体和包尾组成。包头由消息包的长度、目的结点编号、所传消息的编号以及消息内包的编号(包号)等信息组成。包体是真正要传送的数据,包尾则是表示消息包结束的标志信息。

一个完整的消息传递过程包括发送、在互连网络中包的传输和接收3个步骤。第二个步骤主要由硬件完成(参见**路由机制**),其余两个步骤由软、硬件共同完成。消息发送前需将消息分解并按特定格式组成一个个消息包,放入发送结点缓冲区中,然后依次发送到**互连网络**中。目的结点从网络中取包、解包并组成消息。

消息传递中的两个最重要的问题是可靠性和效率。可靠性主要包含3层意思:消息包中个别位出错;消息包传丢或传到非目的结点以及消息传递过程中系统瘫痪。第一种情况可由专门的硬件或软件来发现和纠正。对于第二种情况,可以用适当的消息传递协议来保证将消息包正确地传送到目的结点。但是,这样做可能降低通信效率。第三种情况可能是死锁造成的,也可能是其他软件故障,情况比较复杂。这是大规模并行计算机系统必须解决的一个关键问题。

衡量消息传递效率的一个重要参数是消息包的延迟时间。它包括发送、网络传输和接收3部分时间。对于一个相当规模的**互连网络**(例如10×10的网格),网络传输延迟一般不超过1 μs,而一个发送或接收过程的操作系统额外开销可能超过数十微秒。降低这种额外开销的一个有效办法是使消息包中包含目的地址信息,并在消息传递中尽量减少操作系统的介入。

参考文献

Seitz C L. Concurrent architecture. In: Suaya R, Birtwistle G, eds. VLSI and Parallel Computation. San Mateo: Morgan Kaufmann. 1990, Chapter 3

(祝明发)

xiaoxi chuandi jiekou

消息传递接口(message paasing interface, MPI) 一种与具体语言无关的并行消息通信协议,是一组可移植的,面向高性能计算的消息传递编程接口,支持点对点通信和集合通信,可以基于C, C++和FORTRAN语言通过调用MPI函数开发并行程序。在目前的高性能计算领域,MPI是分布式存储系统上标准并行编程模型。

MPI遵循单程序多数据(single program multi-Data)编程模式,每个并行执行进程有唯一ID,基于自身ID处理不同数据或计算不同任务。进程间需要数据交换时,可以调用丰富的MPI函数进行点对点通信或集合通信。点对点通信是指两个进程间交换数据,一方调用发送函数,另一方调用接收函数。集合通信是指多个进程进行固定模式的数据交换,包括广播、规约、全交换等。MPI在通信时利用通信域(communicator)和标签(tag)来区分不同通信域的无用消息,防止两个进程在多个通信域或者一个通信域内传递多个消息时混淆。为适应不同算法并充分利用通信带宽,MPI还支持自定义通信数据类型。

MPI的标准化始于1992年分布存储环境中消息传递标准研讨会,1993年底推出了MPI 1.0。随后MPI标准由MPI论坛负责,在1995年推出了MPI 1.1版本。1997年,对MPI作了重要扩充,把以前各种版本统称MPI-1,新版本称为MPI-2,加入单边通信函数,动态进程管理和并行I/O(MPI-IO),MPI-2包括200多个函数。

目前流行的MPI实现包括MPICH和OpenMPI。MPICH主要由美国阿贡国家实验室开发和维护,支持InfiniBand的MVAPICH,支持Myrinet的MPICH GM以及Intel MPI都是基于MPICH对不同网络硬件的扩展。OpenMPI由若干早期MPI实现版本合并而成。

参考文献

都志辉. 高性能计算并行编程技术—MPI并行程序设计. 北京:清华大学出版社,2001

(袁良　张云泉)

xiaoxi zhongjianjian

消息中间件(message-oriented middleware, MOM) 利用**消息传递**机制进行平台无关的数据交流,并基于数据通信来进行分布式系统的集成的**分布计算中间件**。

消息中间件适用于需要可靠数据传送的分布式环境。在采用消息中间件的系统中,不同的对象之间通过传递消息来激活对方的事件,完成相应的操作。通常,发送者先将消息发送给中间的消息服务器(程序),消息服务器将消息存放在若干消息队列中,在合适的时候再将消息转发给接收者处理。消息中间件通过提供消息传递和消息排队机制,能够屏蔽掉各种平台及协议之间的特性,支持不同平台的进程之间的通信和应用程序之间的协同。消息中间件能够在客户和服务器之间提供同步或异步的连

接，并且在任何时刻都能传递消息或者进行存储转发，这是它比远程过程调用中间件进步的原因。

消息中间件的核心概念是消息和消息队列。消息是与应用相关的数据结构，包括消息头和消息体。其中，消息头定义了消息本身的一些系统属性和上下文，而消息体的内容则完全与应用相关，称作应用数据。在通信过程中，消息的发送方定义应用数据的格式并提供应用数据，可以是字符、位串、二进制整数和浮点数等所有类型的数据。而消息的接收方也需要了解消息中应用数据的格式，这样才能对接收到的数据进行解释和处理。为了屏蔽异构操作系统和计算机硬件之间数据表示的差异，消息中间件在传递消息时要引入公共数据表示，使得交换信息的双方能够对所交换的信息取得一致的理解。

消息中间件一般采用点对点和发布/订阅等两种传递模型。点对点模型用于消息发送者和消息接收者之间点到点的通信，消息发送者将消息发送到由某个名字标识的特定接收者。这个名字对应于消息服务中的一个队列，在消息传递给接收者之前它被存储在这个队列中。队列可以是持久的，以保证在消息服务出现故障时仍然能够传递消息。**发布-订阅模型**用内容分层结构代替了点对点模型中的唯一目的地，消息发送者发布自己的消息，指出消息描述的是分层结构中的一个主题信息。消息接收者订阅了这个主题，多个应用程序可以就一个主题发布和订阅消息，消息中间件将一个主题已发表的消息路由给该主题的所有订阅者。

参考文献

1. 冯玉琳，黄涛，金蓓弘. 网络分布式计算与软件工程. 2 版. 北京：科学出版社，2011

2. Tanenbaum A S, Van Steen M. Distributed system, principles and paradigms. 2nd ed. 北京：清华大学出版社，2008

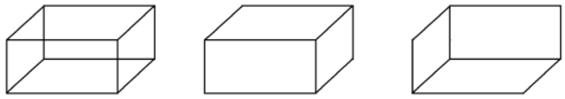
3. Curry E. Message-oriented middleware. In: Middleware for communications. Wiley, 2004: 1-28

（魏峻）

xiaoyin jishu

消隐技术（visibility culling, hidden line/surface removal techniques） 对于三维空间的非透明物体，将观察者所看不见的棱线或表面消除的技术。

为了使计算机生成的图形具有在三维空间的真实感觉，将其看不见的物体部分即隐藏部分消除，这是真实感图形的最基本需求。如果不将隐藏的线或面消除，有时会产生对图形的错误理解。如图 1(a) 是未经消除隐藏线的方块物体，它可以理解成图 1(b) 所示的方块或图 1(c) 所示的方块物体。



(a) 未消隐图　(b) 第一种理解　(c) 第二种理解

图 1　未消隐图形的歧义理解

当空间物体用线框方式（参见**几何造型**）绘制图形时，多面体用棱表示，将观察者所看不见的棱线消除，这时所涉及的消隐问题便是**线消隐**；当物体用具有不同明暗程度的面表示时，将观察者所看不见的面消除，所涉及的消隐问题为**面消隐**。消隐技术从 20 世纪 60 年代即开始进行研究，当时由于图形显示器是线生成器装置，因而消隐问题涉及的主要是线消隐；自从 60 年代后期光栅图形显示器得到广泛应用后，面消隐问题便相应地受到更多关注。线及面消隐技术是计算机真实感图形生成的最基本、最重要的技术之一，多年来研究并实现了许多成功的消隐算法。尽管如此，消隐技术仍然是今天人们所关注的问题。在计算机图形学发展的若干新兴领域，如**多媒体计算技术**、**虚拟现实**等，当实时动态的计算机**真实感图形生成**成为必要时，快速有效地消隐成为十分重要的问题。这时，如何利用计算机软、硬件实现快速的实时消隐成为这些领域的瓶颈技术之一。

线消隐的主要技术有 Roberts 算法等。理论上说，对于物体的每条棱边线段，只要求出它与每个多边形的遮挡关系并进而求出相应的隐藏线段，线消隐问题便可以迎刃而解。然而这种通用的解决办法实际上是难以在实际中应用的，因为它所需要的计算量太大。Roberts 算法针对这一问题采取了有效的解决办法以提高效率。它的核心思想是，根据组成物体多面体的面与面以及面与线之间的相对关系逐步分批地求出隐藏线或隐藏线段。它将所有物体按视线方向从远至近进行排序；从最远的物体开始，对于每个物体，检查和确定与其他每个物体的遮挡关系并计算隐藏线。

面消隐的主要技术有缓冲器算法、扫描线算法以及表优先级算法等。Z-缓冲器算法是利用记录图形深度的缓冲存储器实现面消隐的一种简单实用算法。它在图像空间实施消隐。该算法利用一个记录

投影方向上每个像素上物体投影深度值的缓冲存储器(Z-buffer),每当物体投影时,每个像素上的投影深度将与此像素在Z-缓冲器中已记录的深度进行比较。只有此深度比所记录的深度小时,才将相应点上此物体的颜色或灰度值保存或显示出来。扫描线算法是使用逐行的像素扫描线实行图形绘制和面消隐的一种算法。该算法在图像平面逐行扫描,在每条扫描线上确定和计算可见的扫描线段,即进行消隐处理。扫描线算法还可以与Z-缓冲器算法结合起来。即在每条扫描线上实现Z-缓冲器算法。表优先级算法是按照物体离视点的远近进行优先级排序从而对图形实施有序绘制的一种消隐算法。该算法的处理过程与画家创作一幅油画类似:先画远景,再画中景,最后画近景。因而该算法在处理简单元素(如多边形)时常被称为油画算法。

参考文献

1. 唐泽圣,周嘉玉,李新友. 计算机图形学基础. 北京:清华大学出版社,1995

2. 唐荣锡,汪嘉业,彭群生, 汪国昭,等. 计算机图形学教程. 修订版. 北京:科学出版社,2000

(吴恩华)

xiaoxing jisuanji

小型计算机(mini computer)　位于计算机型谱内最大的多用户系统(**大型计算机**)和最小的单用户系统(**微型计算机**、**个人计算机**)之间的一类多用户**计算机**。在计算机发展历史长河中,由于它具有自己的硬件和**操作系统**,形成了与大型计算机不同特征的一个群体。经典的小型计算机是16位字长计算机,随后出现的更高性能32位小型计算机通常称为**超级小型计算机**(super mini computer)。

发展简史

小型计算机的发展大致可以划分为4个阶段:

第一阶段为20世纪60年代中期,小型计算机开始兴起。这个阶段的特点是采用晶体管或中小规模集成电路的非系列化计算机,字长不统一,软件不共享。一般用于实时过程控制、数据采集和数据处理。典型的产品如1965年DEC公司推出的PDP-8,采用中小规模集成电路,字长12位;IBM公司的用于科学计算的IBM-1130和用于过程控制的IBM-1800。

第二阶段为20世纪70年代,小型计算机开始盛行时期。这个阶段的特点是采用中、大规模集成电路的系列化计算机,机器的性能和软件的功能都有较大的发展,系列化保证了新机型和老机型在软件和外部设备的兼容性。典型的产品有DEC公司的PDP-11系列,其中的PDP-11/70采用高速缓存,在保留单总线结构的同时,与外部设备之间有32位专用数据通路,以确保输入输出的高速度。另外一个较有影响的小型计算机系列是Data General公司的NOVA系列。

第三阶段为20世纪80年代开始的超级小型计算机的鼎盛发展时期。这个阶段的特点是采用中、大规模集成电路,形成了结构开放的硬件和软件标准,开发工具和开发环境成熟以及Unix作为小型计算机操作系统的地位得到确认。典型的产品如VAX-11系列、MICRO VAX系列和王安VS系列等。超级小型计算机的早期典型代表是VAX-11/780,其虚拟地址扩充到32位,采用16个32位通用寄存器;数据类型增加了64位双精度浮点数以及扩展精度等,同时增加了可变位字段数据格式和字符串以及十进制格式。超级小型机VAX-11/780的问世,改变了小型计算机的结构和性能/价格比,使小型计算机得到推广应用。

第四阶段是20世纪90年代,小型计算机开始走向衰落的阶段。随着微处理器的问世和不断发展,促使个人计算机和服务器的兴起,小型计算机时代逐渐结束。

应用和影响

在小型计算机盛行时代,应用领域十分广阔。例如,应用于数据采集和处理、工程设计和计算、信号处理和图像处理、工业过程控制、武器控制、模拟和训练系统、企业管理信息系统以及大型应用系统中的前端处理机、客户-服务器结构中的服务器等。

小型计算机和超级小型计算机在计算机的发展历史上曾有过辉煌时期,发挥过重要的承上启下作用。相对大型计算机而言,小型计算机体积较小、性能价格比好,使计算机走出占据物理空间大的机房而进入部门办公室,由非计算机专业人员使用。微型计算机或个人计算机的诞生也得益于小型计算机的实践。在软件方面,早期微型计算机的操作系统是继承或吸取小型计算机操作系统的经验和理念。例如CP/M类似于RSTS;后来的Windows OS也吸取大量VMS和Unix的长处。1973年出现的Xerox Alto小型计算机,其图形用户界面、数位变换的高分辨率屏幕、鼠标及专门的软件等,为个人计算机的开发迈出具有里程碑作用的一步。　(方全仰)

xie chuliqi

协处理器(coprocessor) 在计算机中,协同主(中央)处理器(CPU)执行某些特定操作,并扩展了CPU功能的专用处理器。

协处理器使计算机在执行这些特定操作的时候,可以获得更好的性能。协处理器是可选件,集成在一个单独的芯片上或与CPU集成在同一个芯片上。协处理器是针对典型应用而设计的专用处理器。典型的例子有浮点运算处理器、**图形处理器**、数字信号处理器、流处理器、多媒体扩展部件等。通过将处理密集型任务从主处理器卸载到协处理器,提高了系统的性能。协处理器扩充了可供程序员使用的**指令系统**,当主处理器接收了一个它不直接支持的操作时,便将控制转移到相应的可以支持该操作的协处理器上。协处理器往往能以很快的速度完成交付给它的任务,使得主处理器可以腾出时间去执行必须由它完成的任务。在一个计算机系统中,可以使用多个协处理器。例如,一个协处理器实现三维图形图像处理,另一个协处理器实现多媒体应用。协处理器方便了专用计算机系列的研制,有助于用于按需配置性能价格比高的计算机系统。

协处理器的概念是从附属处理机发展而来的。附属处理机使计算机系统对不同类型的用户都具有较好的性能价格比。一个计算机可划分成不同的模块,其中,将能满足最大多数用户的常规需求的模块作为主处理机,针对一些特殊用户的需求而设计的专用模块称为附属处理机。这些专用模块在执行特殊用户的特定算法时,可以获得高效率。对于某些特殊的用户,用通用性强的主处理机和特定算法能获得高效率的附属处理机,通过高速通路连接起来,构成适合于这些特殊用户的专用计算机系统。它既具有通用性强的主处理机由于规模生产所带来的低成本的优点,又可用增加不多的代价获得附属处理机在执行该特定算法时带来的高效率。20世纪80年代前,常见的附属处理机是向量处理机。美国的FPS公司曾经是专门生产附属向量处理机的公司,它所生产的AP-120B和FPS-164等附属向量处理机,作为向量部件连接到通用主机(例如DEC公司的超级小型计算机)上,使那些需要进行大量向量计算或矩阵运算的用户获得性能价格比很高的计算机系统。此外,一些著名的计算机公司如IBM、富士通、日立和NEC等,也生产附属向量处理机(或称向量部件或阵列部件)与自己的通用计算机配套。

随着微电子技术和计算机技术的飞速发展,主处理机模块和附属处理机模块可以分别做在超大规模集成电路芯片上,称为**微处理器**和协处理器。跟附属处理机与主处理机系统相比,协处理器与主处理器之间的耦合程度要紧密得多,从而使主处理器管理协处理器所需的系统开销小得多。

协处理器的典型例子是Intel公司的x87系列数值运算协处理器。以Intel 80387数值运算协处理器(以下简称387)为例,它是为Intel 80386微处理器(以下简称386)专门设计的,用来高效执行高精度的浮点运算以及正弦、余弦、正切、对数等超越函数的计算。387协处理器与386微处理器并行连接可以构成一个高性能的386微处理机系统。387协处理器为386微处理器扩充了70多条指令(称为ESC指令,指令的最高5位为11011),还扩充了多种数据类型,如:32位,64位和80位浮点数;32位和64位整数以及18位二进制编码的十进制数,浮点数遵循IEEE 754浮点数标准。387在硬件方面为386扩充了8个80位的浮点寄存器以及16位的控制寄存器、状态寄存器和标志寄存器。系统启动后,387需完成初始化过程,它与386执行同一个指令流,共同完成对指令的译码。当386译码出一条ESC指令后,386将取得的387操作码或387的操作数通过专门为387设定的I/O端口(800 000F8H ~ 800 000FFH)传送给387。接着386与387即可并行工作,386通过BUSY引脚实现与387的同步,通过ERROR引脚检测387需进行的异常处理。这种协同工作方式,大大减少了软件系统的开销。此外,387的所有存储器的访问均由386负责管理。所以,不论386是工作在实地址、保护态或者虚拟8086状态,387都以同样方式工作。387只负责对386传送给它的指令和数据进行操作,不必顾及386的工作模式,从而简化了387的设计。

协处理器的出现是微电子技术和计算机技术发展到一定阶段的产物。随着微电子技术的进一步发展,原来用单独的芯片实现数值协处理器的功能,已经可以与主处理器在同一个芯片中实现。如Intel 80486DX以后的通用微处理器中已包含数值协处理器,于是单独的数值计算协处理器已逐渐消失。随着多媒体应用的发展,几乎所有通用微处理器生产厂商都在其生产的微处理器中集成了一个或多个多媒体扩展部件,该部件挖掘了多媒体应用核心代码的可并行性以及对计算精度要求远低于通用处理器所提供的计算能力的特点。这些多媒体扩展部件通常以通用微处理器已有的计算资源为基础,以向量

部件的形式出现,相应的扩展指令集以单指令多数据(SIMD)的向量指令为主,称为多媒体指令扩展集。图形处理器是另一类热门的协处理器(参见**图形处理器**)。

近年来可编程门阵列(FPGA)芯片其集成度和速度及编程工具有了很大发展,基于FPGA的协处理器体系结构日趋成熟,大大简化了将计算密度型的任务卸载到一个可编程专用硬件处理器的过程。设计者可使用标准化的硬件和软件接口,以及电子设计自动化工具,选择可供某种型号FPGA使用的IP核(参见**知识产权核**)和标准化的应用程序设计接口(API),仅需对硬软件做少量修改,即可实现针对给定应用的FPGA协处理器。该FPGA协处理器可以与现成的主处理器芯片协同工作,也可以与主处理器集成在同一个FPGA芯片中。基于FPGA的协处理器为研制生产小批量、多品种、低成本的协处理器开辟了新的途径,拓宽了协处理器的应用。

总之,随着计算机应用和集成电路技术的飞速发展,协处理器正向着深度和广度发展,并日趋成熟。

参考文献

1. Protopapas D A. Microcomputer hardware design. Englewood Cliffs, NJ: Prentice Hall Inc., 1988

2. Daintith J, Wright E. Dictionary of Computing. 6th ed. New York: Oxford University Press, 2010

3. Intel Corp., Microprocessors Vol. 1. 3065 Bowers Avenue, Santa Clara, CA95051, 1995

(韩承德)

xietong licheng

协同例程(coroutine) 可以和其他例程互相调用的例程,它们彼此处于平等地位,调用后无须返回到开始位置,且自带工作区。

协同例程由两部分组成:不受每次执行影响的固定部分(如机器语言指令)和随不同执行而变的可变部分(如数据和控制信息)。可变部分又称工作区。

协同例程与子例程的区别在于工作区的生存期不依赖于控制进入或离开例程的时间。而且协同例程的工作区需保持局部指令计数器的内容,以便每当再次进入一个例程时,总是从它最近离开的那次执行被停止处开始继续执行。图1中的C1和C2表示两个协同例程。

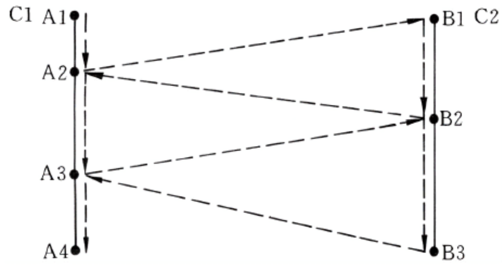


图 1

在一台处理机上,例程C1从起点A1开始,当执行到A2处被中断,欲调用例程C2,就必须先将例程C2的执行点A2的现场保存起来,然后才进入例程C2的起点B1。当C2执行到B2位置又调用C1时,则借助先前保存的现场如A2的位置,继续在曾被中断的执行点A2执行下去,到了A3位置又调用C2。以此类推,直到C1执行到终点A4处为止。

下面介绍一个著名的"快速分类"算法。其思想是:对一组数组元素 $A[m],\cdots,A[n]$,使这些元素分成两部分:对某个合适的j,左部 $A[m],\cdots,A[j-1]$ 包含小于某个值 V 的所有元素,右部 $A[j+1],\cdots,A[n]$ 包含大于 V 的所有那些元素,位于这两部分之间的元素 $A[j]$ 将等于 V。划分进行如下:依次从左到右扫视各个元素,直到找到一个大于 V 的元素为止,这时便从右到左扫视,直到找到一个小于 V 的元素为止。然后,再从左到右扫视,如此等等,每次把错位元素放到相反的一边,直到两种扫视相会时为止。这种算法可以写成一个例程(如引进布尔变量以区分从左扫视与从右扫视),也可写成两个协同例程。可以说从概念上来讲后者较为简单明了,具体程序如下(先赋初值,然后调用 move i):

```
    coroutine move i;   从左到右扫视
        loop: if A[i] > V
    then A[j]: = A[i];
      resume move j;
          fi;
          i: = i + 1;
    while i < j repeat;
coroutine move j;从右到左扫视
    loop: if V > A[j]
          then A[i]: = A[j];
                resume move i;
          fi;
```

```
            j: =j-1;
        while i<j repeat;
        i: =m; j: =n; V: =A[j];
        call move i;
        A[j]: =V;      当 i=j 时
```

其中:语句**resume** move j 在将控制转移到协同例程 move j 之前得先保存 move i 当前执行点的现场。同时,**resume** move i 在将控制转移到协同例程 move i 之前也得先保存 move j 当前执行点的现场。

参考文献

Hansen P B. Operating system principles. Prentice Hall,1973

(段祥)

xindao rongliang

信道容量(channel capacity) 对于给定的信道(通常为通信介质的一部分),存在一种信源(某种输入概率分布),使信道传输的信息量达到最大值,这个最大值称为信道容量,而相应的输入概率分布称为最佳输入分布。信道容量是信道传送信息的最大能力的度量,信道实际传送的信息量必然小于或等于信道容量。

并不是任何信道都有精确的信道容量,对某些信道,只能得到容量的上界和下界,确切容量尚不易规定;对不同信道的信道容量求解,存在较多方法。例如:假设带宽为 B(Hz)的连续信道,其输入信号为 $x(t)$,$x(t)$ 的功率为 S;信道加性高斯白噪声为 $n(t)$,$n(t)$ 的功率为 N,$n(t)$ 的均值为零,则此连续信道的信道容量(单位为比特)为

$$C = B\log_2\left(1+\frac{s}{N}\right)(b/s)$$

上式就是香农(Shannon)信道容量公式,简称香农公式。香农公式表明的是当信号与信道加性高斯白噪声的平均功率给定时,在具有一定频带宽度的信道上,理论上单位时间内可能传输的信息量的最大极限数值(最大传输带宽)。

带宽利用率

带宽利用率(bandwidth utilization)又称实际带宽利用率,是指在特定时刻,信道的实际数据传输带宽与该信道的最大传输带宽的比值。其中,设计最大传输带宽有别于信道容量,该最大带宽取决于所采用的传输技术。在特定时刻的信道实际数据传输带宽也称信道的吞吐率(throughput),其大小与当时的网络状况、用户的多少等多种条件相关,吞吐率是网络设计和用户满意度的重要指标。

例如,在带宽为 100Mbps 的以太网中,由于多用户发送数据包存在碰撞等原因,某一时刻实际的数据传输率为 70Mbps,则带宽利用率为 70%。

参考文献

1. 胡道元. 计算机网络(高级). 北京:清华大学出版社, 1999

2. Stallings W. Data and computer communications. Englewood Cliffs, NJ: Prentice Hall, 1997

(张凌)

xinxi jiansuo fangfa

信息检索方法(information retrieval method) 采集、存储、表示和查询信息的技术手段。其中信息的表示和查询是信息检索方法的关键。信息表示反映计算机看待文本的角度和深度,信息查询要计算文档与查询词的相关度以及文档的重要度。无论是信息表示,还是信息查询,均体现在信息检索模型中。信息检索模型抽象地看待信息内容和检索过程。模型的优劣从根本上决定了检索系统的性能。

迄今提出的信息检索模型主要有布尔检索模型、向量空间模型、概率检索模型、扩展布尔模型、统计语言模型、隐性语义索引模型、基于本体的模型等。

(1) 布尔检索模型提供“与”“或”“非”三种逻辑操作,针对全文检索的需求,又扩充了位置检索功能,如“同一段”“同一句”等。由于布尔模型概念简单,实现容易且检索效率高,因此为商用系统普遍采用。但布尔模型也存在重大缺陷:①命中量很难控制;②难以实现有效的相关排序;③布尔提问式有时会产生违反常规的结果。

(2) 向量空间模型用向量来表示词、文献和提问,通过计算文献与提问所对应的向量之间的相关程度来获得检索结果。Cornell 大学研制的 SMART 系统是向量检索模型的代表。向量检索模型的缺点是计算复杂度很高。

(3) 概率检索模型的基本前提是对于某一给定的提问,在先前检索出的相关文献中出现的词应该较那些未出现的词更为重要。目前已有基于概率模型的实用系统,但概率模型存在参数准确估计的困难。

(4) 扩展布尔模型融合了其他几种模型的优点,通过建立数学模型来表示布尔逻辑,通过加权机制改进索引项的权值,可以实现相关排序。

(5) 统计语言模型试图通过统计学和概率论对

自然语言进行建模,从而获取自然语言中的规律和特性,以解决语言信息处理中的特定问题。基于统计语言模型的检索模型其基本思想是对于每一篇观察到的文档,假设其对应着一个语言模型,并根据这篇观察到的文档估计这个语言模型,从而把信息检索问题转化为对语言模型的估计问题。

(6) 隐性语义模型实际上是将高维空间中的文档向量(索引项向量)投影到低维的潜在语义空间中,将原来在高维空间中比较稀疏的向量压缩到潜在语义空间中,缓解数据稀疏问题。即使原来没有任何共同索引项的两个文档或完全不相同的两个索引项,在隐性语义模型中仍然可能找到它们之间比较有意义的关联值。

(7) 本体论是描述概念和概念之间关系的一种概念模型, 通过概念之间的关系来描述概念的语义, 作为一种有效表现概念层次结构和语义的模型,它能满足用户在语义上和语用知识上的需求,因而在信息检索领域中得到了广泛关注,在很多领域中,基于本体的信息检索模型都取得了很好的效果,大大提高了信息检索系统的准确率和召回率。

参考文献

1. Croft W B, Metzler D, Strohman T. Search engines: information retrieval in practice. London, England: Pearson,2010

2. Salton G. Automatic text processing: the transformation, analysis, and retrieval of information by computer. Addison Wesley, 1989

(刘挺　邵艳秋　施水才)

xinxi jiaohuan bianma

信息交换编码 (information interchange code)　使用一组二进制数字表示字母、阿拉伯数字、基本符号的编码标准。通过编码的数据信息(二进制数据包)可以在数字通信线路上传输,接收端可以解码还原出原始的数据信息。这些字母开始是大小写英文字符(例如 ASCII 编码标准),后来扩展到中文(例如 GB2312 编码标准)和多语种支持(例如 Unicode 编码标准)。

(1) 国际 2 号电报码　国际 2 号电报码又称波多码(Baudot),是一种 5 单位代码。所谓 5 单位是指一个字符可用 5 位二进制码表示,例如字母 A 用代码 11000 表示。国际 2 号电报码是最早采用的一种数据通信字符代码,且目前仍广泛用于电报通信中的电传打字机和电传打印机中,此外,在某些低速数据通信中也使用此代码。

(2) 国际 5 号代码(ASCII 码)　国际 5 号代码是一种 7 单位代码,该代码是美国标准协会于 1963 年提出的,又称美国信息交换用标准代码,即 ASCII 码,后来被国际标准化组织(ISO)和国际电信联盟(ITU)采用,修改并发展成为一种标准的国际通用信息交换用代码,其建议版本号为 ITU-T V. 3。国际 5 号代码是当前数据通信中常用的一种代码,表 1 是其编码表。

ASCII 码表列出每一个字符的十进制值,十六进制值,终端上的显示结果,ASCII 助记名和 ASCII 控制字符定义。

表 1　ASCII 编码表

十进制数值	十六进制值	终端显示	ASCII 助记名	备注
0	00	^@	NUL	空
1	01	^A	SOH	文件头的开始
2	02	^B	STX	文本的开始
3	03	^C	ETX	文本的结束
4	04	^D	EOT	传输的结束
5	05	^E	ENQ	询问
6	06	^F	ACK	确认
7	07	^G	BEL	响铃
8	08	^H	BS	后退
9	09	^I	HT	水平跳格
10	0A	^J	LF	换行

续表

十进制数值	十六进制值	终端显示	ASCII 助记名	备注
11	0B	^K	VT	垂直跳格
12	0C	^L	FF	格式馈给
13	0D	^M	CR	回车
14	0E	^N	SO	向外移出
15	0F	^O	SI	向内移入
16	10	^P	DLE	数据传送换码
17	11	^Q	DC1	设备控制 1
18	12	^R	DC2	设备控制 2
19	13	^S	DC3	设备控制 3
20	14	^T	DC4	设备控制 4
21	15	^U	NAK	否定
22	16	^V	SYN	同步空闲
23	17	^W	ETB	传输块结束
24	18	^X	CAN	取消
25	19	^Y	EM	媒体结束
26	1A	^Z	SUB	减
27	1B	^[	ESC	退出
28	1C	^*	FS	域分隔符
29	1D	^]	GS	组分隔符
30	1E	^^	RS	记录分隔符
31	1F	^_	US	单元分隔符
32	20	(Space)	Space	
33	21	|	|	
34	22	`	`	
35	23	#	#	
36	24	$		
37	25	%		
38	26	&		
39	27	‘		
40	28	(		
41	29	)		
42	2A	*		
43	2B	+		
44	2C	,		
45	2D	-		

续表

十进制数值	十六进制值	终端显示	ASCII 助记名	备注
46	2E	.		
47	2F	／		
48	30	0		
49	31	1		
50	32	2		
51	33	3		
52	34	4		
53	35	5		
54	36	6		
55	37	7		
56	38	8		
57	39	9		
58	3A	：		
59	3B	；		
60	3C	<		
61	3D	=		
62	3E	?		
63	3F	?		
64	40	@		
65	41	A		
66	42	B		
67	43	C		
68	44	D		
69	45	E		
70	46	F		
71	47	G		
72	48	H		
73	49	I		
74	4A	J		
75	4B	K		
76	4C	L		
77	4D	M		
78	4E	N		
79	4F	O		
80	50	P		

续表

十进制数值	十六进制值	终端显示	ASCII 助记名	备注
81	51	Q		
82	52	R		
83	53	S		
84	54	T		
85	55	U		
86	56	V		
87	57	W		
88	58	X		
89	59	Y		
90	5A	Z		
91	5B	[		
92	5C	“		
93	5D	]		
94	5E	ˆ		
95	5F	_		
96	60	‘		
97	61	a		
98	62	b		
99	63	c		
100	64	d		
101	65	e		
102	66	f		
103	67	g		
104	68	h		
105	69	i		
106	6A	j		
107	6B	k		
108	6C	l		
109	6D	m		
110	6E	n		
111	6F	o		
112	70	p		
113	71	q		
114	72	r		
115	73	s		

续表

十进制数值	十六进制值	终端显示	ASCII 助记名	备注
116	74	t		
117	75	u		
118	76	v		
119	77	w		
120	78	x		
121	79	y		
122	7A	z		
123	7B	{		
124	7C	|		
125	7D	}		
126	7E			
127	7F		DEL	Delete

ASCII 码虽然只有 7 位二进制，但为了传输的可靠性，常在码字的尾部增加一个第 8 位以做奇偶校验用，这样可由在该 8 位二进制码组中出现“1”的总个数为奇数或偶数来判定是否在传输中产生错误，以便采取相应的措施。

（3）汉字编码　汉字代码是汉字信息交换用的标准代码，适用于一般的汉字处理和汉字通信系统之间的信息交换，国际代号为 GB 2312—80，其基本点是对于任何一个汉字字符均采用两个字节表示。

（4）Unicode 编码　Unicode 是目前计算机学术和产业界认可的一项通用编码方案。Unicode 可以支持世界上大部分文字系统的统一编码，第六版已经收入的字符超过十万个。广泛用于互联网领域的扩展标记语言 XML 及其子集 HTML 就采用 UTF-8 作为标准字符集，Unicode 不但用于信息交换，也逐步用于新的程序设计语言和新的操作系统。

参考文献

刘云，刘志华，郑宏云，张振江．计算机网络实用教程．2 版．北京：清华大学出版社，2006

（周杰　张凌）

xinxi keshihua

信息可视化（information visualization）　研究如何将大规模抽象数据转换为图形图像，并利用计算机交互来帮助人们理解和展示数据的一门学科。信息可视化兴起于 20 世纪 90 年代，是**可视化**的主要分支之一。相对于另一主要分支科学可视化，信息可视化处理的数据主要是所谓的抽象数据，即数据本身并没有内在的二维或三维几何结构，比如文本数据、社交网络、软件代码等。信息可视化是一门交叉学科，涉及**人机交互技术**、**计算机图形学**、心理学、平面设计等。

信息可视化的主要研究内容包括：有结构数据的可视化、无结构数据的可视化、可视化技术的评估和比较以及交互技术。

有结构数据的可视化包括：①时间序列的可视化。时间序列一般表示为一维有序线性表，表中的每一个数据实体按照其相关的时间进行排列。代表性的时间序列可视化技术有（line chart），时间链（time line）等。②多维数据的可视化。多维数据一般表示为抽象数据表。表的每一行代表一个数据实体，每一列代表实体的相关属性。多维数据可视化旨在解决如何有效并且直观的将多维信息映射到低维可视化元素（例如点、线等）或可视化属性（例如颜色、形状、大小、面积等）之上的问题。比较有代表性的多维数据可视化技术有平行坐标系（parallel coordinates）、散点图矩阵（scatter plot matrix）等。③各种关系的可视化。其中包括树的可视化以及图的可视化。树的可视化关注于层次关系（hierarchical relation）的展示，例如组织结构树（organization tree）及树状矩形图（treemap）等。图的可视化，作为可视化领域的核心问题之一，主要用于展示数据元素之间的复杂关联以及拓扑结构。尤其是当图的规模很大，或是随时

间变化，或是附带多维属性时图的可视化能够帮助用户更好地获取数据中的关联信息。④地理信息的可视化。用于展现真实地理数据以及伴随与地理数据值上的信息（如人口密度、交通流量等）。

无结构数据的可视化包括：①文本信息的可视化。其中包括对文本主题的可视化，对文本语义关联的可视化，以及对文本观点及情感的可视化。这些技术与文本数据挖掘紧密相关，往往起到展示文本分析结果、诠释文本信息内涵的目的。典型的技术有主题流（theme river）、短语图（phrase net）以及观点三角形（opinion triangle）等。②多媒体信息（图像，音频，视频）的可视化。其中包括对媒体信息特征的可视化以及对媒体文件关联的可视化。对媒体信息特征的可视化往往用于展示某单一媒体文件内容的特征。典型的技术有音乐可视化（music visualization），用于展现一支音乐在播放过程中的频谱、响度以及音调等特征的实时变化。语义图像浏览器（semantic image browser），用于展现图片的语义特征。媒体文件关联的可视化往往被用来展示媒体文件的关联，从而达到帮助信息归类、方便信息查询的目的。典型的技术包括将用图的可视化技术来展现视频或图像之间的关联。

如何评估可视化技术的有效性以及比较不同的可视化技术是信息可视化领域的难点。现有的可视化评估方法往往借助于用户调研技术。常用的技术包括主观评估以及客观评估。主观评估通过问卷调查或直接交流等方式来获得用户对可视化工具的直观性、交互性等的反馈；客观评估通过要求用户使用可视化工具完成预定任务，记录并分析相关的定量数据如完成任务的时间、正确率等，从而评估系统的有用性、易用性及高效性。

交互技术。动态交互将信息可视化工具与传统的静态信息展示工具区分开来。信息可视化工具中的常见交互技术包括，动态查询（dynamic query）、关联性刷选（brushing and brushing）、语义变焦（semantic zoom）、整体与细节变换（overview + details）以及焦点与环境变换（focus + context）等。动画往往伴随于交互技术，使数据变化及信息的切换以连贯光滑的形式展现于用户面前，从而避免了视觉上的间隙。交互技术使得信息可视化工具被广泛地应用于数据探析（explorative analysis）当中。

信息可视化技术有着广泛的应用。究其应用领域，信息可视化可以分为金融数据的可视化、社交媒体的可视化、新闻的可视化、软件程序的可视化、生物医学和生物信息的可视等。

当今，海量数据的可获取性导致信息爆炸。如何帮助用户在纷繁复杂的数据当中找到并理解有用信息是信息科学所面临的共同难题。信息检索技术帮助用户快速找到相关数据；**数据挖掘**技术帮助用户分析潜在的数据模式；而信息可视化技术，旨在帮助用户高效地理解原始数据及数据分析结果以获取有用信息。具体而言，信息可视化可以在以下三个方向上推动信息技术的革新：

（1）与搜索的紧密结合　传统的信息检索工具往往将搜索结果按检索相关度排列并以列表的形式加以展示，缺乏对于信息整体的总结及概括。信息可视化技术利用人类与生俱来的图形模式识别能力，能以较为紧凑的空间总结并概括大量信息。从而大大提高用户对搜索信息筛选及知识获取的效率。

（2）面向大众的可视化应用　众多先进的可视化技术仍然缺乏推广及真实应用的检验。如何让没有专业背景的大众也可以使用可视化技术来理解和表现他们的数据有着广阔的前景。

（3）与数据分析结合　信息可视化和分析紧密结合形成可视化一个新的分支——可视分析。可视分析侧重于理解、分析、决策，已经成为另一个研究热点。

参考文献

1. Card S K, Mackinlay J D, Shneiderman B. Readings in information visualization: using vision to think. San Frandcisco, CA: Morgan Kaufmann Publishers, 1999

2. Ware C. Information visualization: perception for design. 3rd ed. San Francisco, CA: Morgan Kaufmann, 2012

（屈华民）

xinxi tiqu

信息提取（information extraction）　利用自然语言处理和机器学习等方法从非结构化（或半结构化）的数据中提取出结构化信息的过程。广义的信息提取涉及的数据包括文本、视频、音频等，随着研究领域的细化，目前人们所说的信息提取一般仅限于基于文本处理的过程。由于大量的有用信息都是以自然语言的形式存在于文本当中，一方面不便于计算机的理解，另一方面存在大量噪声数据，因此信息提取方法旨在提供更有力的信息获取工具，增强计算机对文本的解读能力和处理大规模文本的能

力。例如，同一主题的信息通常分散在不同的网页中，表现形式也不尽相同（包括文本、链接、表格等），若能将这些信息提取整合，以结构化的形式储存管理，则既方便了用户查询，也为其他与数据挖掘相关的应用提供了有价值的信息。

发展简史　信息提取的历史可以追溯到20世纪70年代，与早期的自然语言处理技术同时出现。从1987年开始举办的消息理解会议（Message Understanding Conferences，MUC）以及随后出现的自动内容提取会议（Automatic Content Extraction，ACE）、多语种实体任务评价（Multilingual Entity Task Evaluation，NET）和文本分析会议（Text Analysis Conference，TAC）均推动了信息提取技术的发展。这些会议都是以竞赛的形式举办，具体任务包括：命名实体识别、多语言实体识别、关系提取、共指消解等。会议组织者会为各种竞赛任务提供人工标注完成的数据集供参会者使用，最终大会对所提交的解决方案进行评测。随着各项任务相关技术的不断融合，目前的信息提取技术已经成为自然语言处理领域的一个重要分支。

关键技术及典型任务　信息提取涉及的关键技术包括自然语言处理、机器学习和统计分析等。总体说来，信息提取的方法分为监督式和非监督式两大类，主要包括基于模式的方法、基于分类器的方法和序列模型的方法等。近来，基于序列模型的方法以其较高的准确率被广泛地应用于信息提取系统中，其所用到的序列模型主要包括隐马尔可夫模型、条件随机场模型和马尔可夫逻辑网络等。典型的信息提取任务包括：命名实体识别（entity recognition）、实体解析（entity resolution）、关系提取（relation extraction）、概念提取（concept extraction）、事件发现与识别（event discovery and identification）和总结提取（summary extraction）等。

重点应用领域及典型应用　随着万维网的发展，非结构化的文本呈指数级增长，人们迫切地需要信息提取系统帮助处理如此庞大规模的在线数据，将蕴含在文本中的有用信息转化为结构化数据，以便于计算机的有效处理。因此，信息提取在万维网如此发达的今天发挥着至关重要的作用。知识库的构建和扩展是语义网发展的基础，而从万维网中通过信息提取得到的结构化数据对于大规模知识库的构建和扩展都非常重要。同时信息提取技术有着大量的实际应用，例如：网页中的产品信息整合、人名排歧系统、问答系统、实体关系搜索系统等。

评价标准　信息提取系统的评价标准主要有查准率（precision）和查全率（recall）。查准率重在评价提取信息中正确提取的比例，查全率重在检验正确提取的信息占所有正确信息的比例。在对两个信息提取系统进行性能对比的时候，一般使用这两个评价标准的调和平均值$F1$. 如以下公式所示

$$F1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

面临的挑战及未来发展方向　在信息社会高度发达的今天，信息提取工作面临着更多的挑战。对于提取指定类型信息，例如人名、地名和时间等，已存在很多提取模型，但它们大多是通过对新闻网页或者指定领域的文本进行训练而产生的。对于上下文日益复杂的互联网环境，要同时保证信息提取的准确性和查全率，仍需要更多的努力。另外，由于万维网数据的异构性和多样性，除了对给定类型信息的提取，人们同时希望自动提取出蕴含在文本中的未知的有用信息。

现在的信息提取系统主要处理万维网中的非结构化文本，虽然取得了一定的成功，但是还存在很多待解决的问题，例如，跨文本信息提取，跨语言信息提取和域自动适应性等。近年来，非文本文件的信息提取越来越引起研究者的关注，从音频、视频等多媒体文件中提取出结构化的信息将成为未来信息提取的重要方向。此外，除了信息本身的提取，不同信息之间关系的提取也是近几年的热点研究问题。

参考文献

1. Manning C D. 统计自然语言处理基础. 苑春法，等译. 北京：电子工业出版社，2005

2. 程显毅，等. 中文信息抽取原理及应用. 北京：科学出版社，2010

3. Moens M-F. Information extraction：algorithms and prospects in a retrieval context. Springer，2010

（王建勇）

xinxi yinbi

信息隐蔽（information hiding）　在大型程序设计中，为了实现对象的可见性控制，在分层构造软件模块时要求有些对象只在模块内部可见，在该模块外部不可见的软件开发的原则和方法。例如在自顶向下分层设计中，其较低层的设计细节都被“隐蔽”起来，不仅功能的执行机制被隐蔽起来，而且控制流程的细节和一些数据也被隐蔽起来，随着设计逐步往低层推移，其细节也逐步显露出来。

在模块化设计中,接口只是功能描述,而模块本身的实现细节对外界则是不可见的,实现上的改变并不影响使用它的模块,这有利于软件的重复使用。

例如,在 Ada 语言中,通过把包块分为两部分,使得外部元素和隐蔽元素明显地分开,包块式(又称包块规约)只说明外部元素,包块体则包括了隐蔽元素的说明,隐蔽元素只是支持外部元素的实现。

参考文献

徐家福. 系统程序设计语言. 北京: 科学出版社,1983

(程虎)

xinyuan jiaohuan

信元交换(cell switching) 以称为信元的定长分组为数据传输单位的一种交换技术。又称信元中继,它是异步传送模式(ATM)采用的交换技术,目前已经被采用 TCP/IP 技术的互联网所取代。

(史美林 徐明伟)

xingshi fangfa

形式方法(formal method) 建立在严格数学基础上的软件开发方法。在软件开发的全过程中,从需求分析、规约、设计、编程、系统集成、测试、文档生成、直至维护的各个阶段,凡是采用严格的数学语言,具有精确语义的方法,都属于形式方法。

在软件设计与研制中采用形式方法,可追溯到20世纪50年代后期对于**程序设计语言**编译技术的研究。那时 J. Backus 提出了 BNF 记法作为描述程序设计语言语法的**元语言**,出现了各种语法分析程序自动生成器以及语法制导的编译方法,使编译系统的开发从"手工艺制作方式"发展成具有牢靠理论基础的系统方法。

形式方法的研究真正形成热潮是在60年代后期。面对当时出现的所谓"软件危机",人们提出了种种解决方法,归纳起来有两类:一是采用工程方法来组织、管理软件的开发过程,二是深入探讨程序和程序开发过程的规律,建立严密的理论,以期能用来指导软件开发实践。前者导致"软件工程"的出现和发展,后者则推动了对形式方法的深入研究。

形式方法所要解决的核心问题是程序的正确性。当今的软件系统极为庞大且高度复杂,不可能只靠测试来保证其正确性(荷兰的计算机科学家 E. W. Dijkstra 曾经有一句名言:"测试只能表明程序中存在错误,而不能表明程序中没有错误")。因此,应用数学工具来加深对程序和程序开发过程的理解,是从根本上解决程序正确性问题的一个重要途径。

形式方法的一个重要研究内容是**形式规约**,即用具有精确语义的形式语言书写的程序功能描述。形式规约之所以重要,是因为它是设计和编制程序的出发点,也是论证程序是否正确的依据。程序的正确性是一个相对的概念。说一个程序是否正确,总是相对于某个给定的规约而言。如果规约本身是非形式的、不精确的,就无从谈起程序的正确性。

一旦给出系统的形式规约,就可以推断它是否具有某种性质(比如不会发生死锁)。这些性质虽不是规约的一部分,但由于规约是形式的,所以可以在相应的形式推理系统中进行推导。如果用 SPEC, P 和 SYS 分别表示规约、性质和程序,用 $\vdash$ 和 $\models$ 分别表示推导和满足关系,并假定所用的推理系统是可靠的,那么从 SPEC $\vdash$ P 与 SYS $\models$ SPEC 就知道 SYS $\models$ P。换句话说,只要规约具有性质 P,则任何正确地实现了该规约的程序都满足 P。注意这种推理与具体实现无关,可在动手编写程序之前进行。如果发现问题只须修改规约,从而避免了代价高昂的事后修改程序。

程序验证可以说是形式规约的孪生兄弟。事实上,能像证明数学定理那样,形式地证明程序的正确性,是提出形式规约的主要目的之一。每一种形式规约方法均伴随着相应的程序验证方法。比如用于推导命令式程序正确性的 Hoare 逻辑是建立在前、后断言规约方法上的;用于验证并发系统是否具有所希望性质的模型检查方法则基于时序逻辑。

围绕各种程序验证方法出现了大量的程序验证工具,既有自动的,也有交互式的。自动验证工具的好处是不需要人的参与,只要输入规约和程序,系统按照一定的算法,检查程序是否满足规约;在不满足时,大多数系统能给出诊断信息,指出不满足的原因。自动工具的局限性在于所能处理的问题类受限制。自动工具所能对付的问题必须是可判定的;即使对可判定的问题,如果时间、空间复杂性太高,在实际中也是不可行的。在交互式验证系统中,整个证明过程是在人的指导下进行的。系统可随时求助于人的智慧来决定下一步怎么做,因此从理论上说可以对付任何问题。但过多地要求人的干预会导致证明效率低下等问题。实际上大多数交互式工具都含有不同程度的自动化成分。

通过事后验证来保证程序的正确性，显得过于消极。不但难以措手，而且为时已晚。较为积极的办法是研究程序开发过程本身，探寻保证正确性的程序开发方法。这些方法一般是“目标制导”的，即从给定的形式规约出发，应用适当的规则，逐步推出可执行的程序。由于这些规则是保持正确性的，所得到的程序一定满足所给的规约，而无须事后另作验证。这样的程序推导（或转换）涉及不同层次上，即从规约到程序的各种对象，因而要求对所用的规约语言与编程语言的语义在同一框架中给出。例如在“最弱前置条件演算”中，规约断言与程序语句（“卫式命令”）的语义都统一用最弱前置条件来定义。程序推导与转换在函数式与逻辑式程序设计风范中也得到深入的研究。

在直觉主义逻辑中，“证明”必须是能行的，利用最近20年来直觉主义类型论的成果，可以将类型论中的“谓词”看作对程序的规约，谓词为真的证明即是（带类型的）λ-演算的项，因而可看作是函数式程序语言中的程序。这样，“程序满足规约”这句话翻译为类型论的语言就是“论据证明谓词为真”。基于这种对应关系，就可以在直觉主义逻辑系统中进行证明，而从证明中“抽取”出程序（参见**基于类型理论的方法**）。

近年来对实时系统形式方法的研究取得了很大进展。在实时系统中时间因素对系统的行为起关键的甚至是决定性的作用。典型的实时系统有飞机或导弹的飞行控制系统、各类工业过程控制器以及通信系统等。目前比较成功的用于实时系统的方法，多数是在已有的关于非实时系统的方法基础上扩充时间因素，比如在时序逻辑（线性或区间）中引入时间参数，或在传统的进程代数中增加表示时间的算子。一般要求这种扩充是保守的，即在原形式系统中为真的事实在扩充后的系统中仍然为真。**混成系统**是一类特殊的实时系统，其特点是既随时间而连续变化，又受离散突发事件的驱动。对混成系统形式方法的研究方兴未艾。

形式方法的另一个重要应用领域是系统的分解与集成。大型软件系统由成千上万个模块组成，这些模块又按一定的原则组织成一些子系统。模块和子系统的对外界面与版本信息，以及模块、子系统之间的调用、联结关系，都可以形式地加以描述。这种系统结构描述一般在系统设计阶段进行。在系统集成阶段可以利用这种结构描述进行界面与版本一致性检查，也可以自动地从模块库产生最终的可运行系统。实际上，几乎所有的软件开发环境都不同程度地提供这方面的支持。

形式方法的研究用到逻辑、代数、范畴论诸数学分支的成果，同时反过来也推动了这些数学分支的发展。比如λ-演算、时序逻辑、直觉主义类型论近30年来取得的重大进展，都与形式方法有直接的关系。

经过近30年的大量、艰辛的研究，人们在形式方法这一领域已经取得了许多重要的成果，并应用于实际。比如对于抽象数据类型的讨论为高级语言中模块构造的设计提供了理论基础，也是产生面向对象程序设计风范的源头之一；对于通信并发系统的理论研究，导致了用于书写通信协议规约的国际标准语言 Estelle 和 Lotos 的诞生；程序验证系统查出了软件和硬件设计中的许多错误和隐患；现有的自动验证技术已能对付具有 10^{200} 以上状态的系统。但从总体看，迄今为止形式方法所能处理的问题的规模还嫌太小。如何将已经成功地应用于中小规模问题的方法“放大”以解决真正工业规模的问题，是形式方法当前所面临的一个重大挑战。

参考文献

1. Wing J M. A specifier's introduction to formal methods. IEEE Computer, 1990,23(9): 8-24

2. Leveson N G. Special issue on formal methods in software engineering. IEEE Transactions on Software Engineering,1990,16(9)

（林惠民）

xingshi guiyue

形式规约（formal specification） 用具有坚实数学基础的语言和方法给出的软件的功能描述或设计描述。软件的规约既指对最终产品的功能描述、实时特性、运行效率及安装与运行的时空要求等，也指对开发过程的各种要求，比如交货期限、工程预算等，亦可指软件设计描述。它是软件设计与研制工作的出发点，也是衡量最终产品是否合格的依据。形式规约只关心软件产品的功能性质，即“做什么”，而不关心其功能是怎样实现的。

形式规约的最主要特点是具有严格的数学基础，因此，有可能讨论它的一致性与完备性等性质。**一致性**指的是自身无矛盾，这是任何规约都应满足的基本要求。自相矛盾的规约在理论上是没有意义的，在实践上是有害的。因为它将导致人力与财力的浪费。完备性是指规约是否完全、无遗漏地刻画了它所要描述的对象。完备的规约具有许多良好的

性质,比如语义模型的唯一性等。

形式规约的一个重要应用是程序正确性验证。只要所用的编程语言同样具有严格的数学语义,就可以形式地论证程序是否满足所给的规约。如果满足,就说明程序具有规约所刻画的性质。也就是说,程序相对于该规约是正确的。验证程序的正确性需要进行大量的推理和计算,因此,出现了许多支持程序正确性验证的软件工具,即程序证明器。这些工具既有自动的,也有交互式的。

按所依据的数学理论,形式规约可以分为基于集合论的,基于逻辑的与基于代数的;按所适用的程序类,又可以分为面向顺序程序的与面向并发程序的。以下对常用的形式规约方法作简要的介绍。

前后断言方法可以说是最早的形式规约方法,出现于20世纪60年代后期,首倡者是C. A. R. Hoare。其基本思想是在程序语句间插入刻画程序状态(即程序变量的取值)的一阶谓词公式,称为*断言*。插在语句之前的断言指明执行此语句之前程序变量所应具有的性质,称为前断言;插在语句之后的断言则刻画了语句执行结束时的程序状态空间,称为后断言。后断言可以看作是对程序所应实现的任务的描述;前断言则可看作是正确执行程序的前提。一对前、后断言即是对程序的规约。如果用S,P和Q分别表示程序、前断言和后断言,则程序S满足由P,Q组成的规约这一事实可用三元组P{S}Q来表示。C. A. R. Hoare提出了一套公理系统用于推导这样的三元组,如果P{S}Q可由该系统推出,则S就是相对于P,Q正确的。考虑到程序是否终止,这种正确性又可分为部分正确性与完全正确性两类。

形式规约是对程序“做什么”的数学描述,这种描述主要体现在后断言中,因此后断言往往包含了关于程序结构的重要信息。基于这种认识,E. W. Dijkstra提出了一个推导正确程序的形式系统,称为*最弱前置条件演算*。其基本思想是将程序设计看作是“面向目标”的活动,编程就是从预先给定的后断言出发,逐步推导出满足它的程序,同时计算出所需的最弱前置条件。

前、后断言法主要面向程序的控制结构,而程序是控制结构和数据结构的统一体。这一事实在70年代初兴起的研究抽象数据类型的热潮中得到了生动的反映。抽象数据类型由一组数据和施加于该数据上的运算组成。对于这组数据的操作只能通过在该抽象数据类型中定义的运算进行,从而达到了数据抽象和信息隐藏的目的。

为了将前、后断言法推广应用于抽象数据类型的规约,需要引入“表示类型”与“数据不变式”的概念。所谓“表示类型”即在规约中用来表示抽象类型的具体数据类型。比如先进后出栈可用数组加指针来表示。作为表示类型的数据类型的语义必须是已知的;它们可以是在规约语言中预定义的类型,也可以是已经给出了形式规约的其他抽象数据类型。由于抽象数据类型的数据空间常常只是表示类型的一个子集,需要用一谓词来对表示类型加以限定,称为数据不变式。表示类型的元素中只有满足数据不变式的才是抽象类型的合法元素。给定表示类型与数据不变式后,抽象类型上的运算可用前后断言来刻画,这些断言中可引用表示类型上的运算。这种方法称为“面向模型的方法”,主要用在VDM和Z中。

抽象数据类型的另一种(也是更为流行的)规约方法是代数方法,通常称为代数规约。其基本出发点是用基调给出抽象数据类型各运算的类型,而用一组代数公理来规定这些运算所应具有的行为。代数规约的语义即是满足该公理集的某类模型(参见**代数规约**)。代数方法的长处一是数学基础严密,建立在传统的泛代数理论上;二是抽象程度高,只涉及性质,无须诉诸数据的表示,因此又称为面向性质的方法。

在书写规约时应注意的一个问题是如何描述得恰如其分,既不过多也不过少。在规约中描述过多会导致“实现偏向”,给实现施加了不必要的限制,从而排除了一些原本是合理的实现;描述得过少又有容纳不合理实现的危险。为了使规约写得恰到好处,除了应透彻理解、熟练掌握所使用的规约语言和方法外,更重要的是对所要描述的系统有全面深入的了解。

近年来有些学者研究基于直觉主义类型理论的形式规约方法。这种方法将规约看成谓词(类型),将实现规约的程序看成该谓词为真的证明。其特点是可从规约的正确性证明中抽取出可执行程序,这个程序就是该规约的一个实现。

并发程序比顺序程序表现出更为复杂的行为,对它们的规约也更为困难。

时序逻辑是描述并发程序性质的一个有力工具。并发程序的一些重要性质,比如活性(“好的”事件终将发生)、安全性(“坏的”事件永远不会发生)都可以用时序逻辑公式表达。通过在程序中插时序断言,可用与前后断言法类似的方法来证明一

个并发程序满足某时序逻辑公式。对于有限状态的并发系统还可以用模型检查算法来自动判定是否具有用某类时序逻辑公式表达的性质。

70 年代末英国学者 C. A. R. Hoare 与 R. Milner 分别提出了用于描述通信并发系统的代数语言 CSP 与 CCS,后来发展成进程代数理论。进程代数研究的核心问题之一是等价性,即如何决定两个语法上不同的进程表达式表示同一个语义对象。如果用某进程表达式作为一并发系统的规约,则另一与之等价的且包含更多并发性的表达式就可以看作是该规约的实现。进程代数理论提供了多种方法来论证这种实现的正确性。

形式规约研究的重要内容之一是形式规约语言,即用于书写形式规约的语言。不同的形式规约方法要求不同的形式规约语言。比如 Z 和 VDM 各有其规约语言。代数规约语言就更多了,如 Clear, ASL, ACT-ONE /TWO, OBJ。还有面向分布式系统的 Estelle 和 Lotos 等。由于所依据的数学理论及规约方法不同,各种规约语言千差万别。但从结构上看,却具有一个显著的共性:每个规约语言都由两部分组成:描写基本规约(或原子规约)的成分和把基本规约组合成大规约的构造成分。其中后者是形式规约语言研究和设计的重点,也是衡量规约语言优劣的主要依据。常见的构造成分有:包含、并、交、扩充、参数化等。

在整个软件设计和开发过程中,形式规约是从非形式化的、不精确的现实世界到形式化的、精确的程序世界的分界岭。如何从非形式需求得到形式化的规约?这就是形式规约的获取问题,是这一领域中的一个重要研究课题。

参考文献

Wing J M. A specifier's introduction to formal methods. IEEE Computer, 1990,23(9):8-24

(林惠民)

xingshihua yanzheng

形式化验证(formal verification) 使用基于数学的形式化方法来检验一个计算机系统是否具有某种性质,或者是否满足某个形式化规约。在验证的过程中,人们首先需要使用逻辑公式来描述待验证的性质,并给出系统的数学模型或者形式语义。以此为基础,人们可以用数学方法来验证系统或系统模型是否满足待验证的性质。

形式化验证的方法可以分成两大类:模型检验和定理证明。

模型检验的基本思想是通过对系统模型的状态空间进行穷尽搜索,验证系统是否满足特定的性质。这类技术通常要求系统模型的状态空间是有穷的,或者可以通过某些抽象方法转换成有穷的。被检验的性质通常使用时序逻辑公式来描述,有时也会被简化成为特定(关键)状态的可达性。系统模型的描述方法通常包含自动机、转换系统、Petri 网等。对于实时系统,人们通常会在这些模型上添加时钟机制来描述时间特性。

模型检验技术通常被用于验证并发系统、协议等。当模型的规模增大时,状态空间的大小会呈现爆炸性增长,使得对空间的穷尽搜索变得不可能实现。为此,人们设计了很多技术来提高模型检验算法的效率,包括:高效的符号化表示、偏序约减等技术。模型检验通常只能针对系统模型进行验证。它不能够发现实现阶段引入的错误。如果系统的实现和模型不一致,那么系统仍然会出错。

另一种形式化证明方法是定理证明,或者说逻辑推理。这种方法首先给出系统描述语言的形式语义,然后给出一系列的推导规则。系统的性质也通过逻辑公式给出。形式化验证过程通过逻辑推理,给出系统性质的数学证明。也就是说,系统的正确性实际上是这个证明系统的一个定理。定理证明的过程非常繁复和琐碎,本身很容易出错。因此,定理证明过程通常需要借助一些定理证明系统(比如 PVS, Coq)来完成。虽然这类证明系统具有强大的功能,程序的证明仍然要求使用者根据他对程序的理解来完成。因为这一类证明技术要求使用者对逻辑公式和定理证明工具非常熟悉,同时理解程序的工作原理,因此很难应用到普通软件的开发过程中。

在工业界,形式化验证技术被广泛应用于硬件设计中,人们会使用形式化方法来验证集成电路的很多重要性质。但是形式化验证技术在软件开发中的应用较少,主要原因可能是因为软件的灵活性和软件需求的变化性。

参考文献

1. Clarke Jr E M, Grumberg O, Peled D A. Model checking. Cambridge, MA: MIT Press, 1999

2. Hoare C A R. An axiomatic basis for computer programming. Communications of the ACM, 1969, 12(1)

3. Floyd R W. Assigning meaning to programs. In: Mathematical Aspects of Computer Science, Proceedings of Symposium in Applied Mathematics,

Vol. 19, A. M. S., 1967: 19-32 (赵建华)

xingshi yuyan lilun

形式语言理论(formal language theory)

用数学方法研究自然语言(如英语)和人工语言(如程序设计语言)的语法的理论。它只研究语言的组成规则,不研究语言的含义。

形式语言理论在自然语言的理解和翻译、计算机语言的描述和编译、社会和自然现象的模拟、语法制导的模式识别等方面有广泛的应用。

形式语言的研究始于20世纪初,50年代中期将形式语言用于描述自然语言。1956年,N. Chomsky发表了用形式语言方法研究自然语言的第一篇文章。他对语言的定义方法是:给定一组符号(一般是有限多个),称为字母表,以Σ表之。又以Σ^*表示由Σ中字母组成的所有有限长符号串(或称字,包括空字,也称句子)的集合,则Σ^*的每个子集都是Σ上的一个语言。例如,若令Σ为26个拉丁字母加上空格和标点符号,则每个英语句子都是Σ^*中的一个元素,所有合法的英语句子的集合是Σ上的一个语言。乔姆斯基的语言定义方法为人们所公认,一直沿用下来。

1960年,算法语言ALGOL 60报告发表,其中第一次使用一种称为巴克斯范式的方法来描述程序设计语言的语法。不久,人们即发现巴克斯范式系统极其类似于形式语言理论中的上下文无关文法,从而打开了形式语言广泛应用于描述程序设计语言的局面,使它发展成为理论计算机科学的一个重要分支。

形式语言和文法 形式语言理论以无限的语言为主要研究对象。例如,所有以n个a构成的字($n \geqslant 1$)组成一个语言$L_a = \{a, aa, aaa, \cdots\}$,它就是无限的。因此,研究形式语言遇到的第一个问题就是描述问题。描述的手段必须是严格的,而且必须能以有限的手段描述无限的语言。乔姆斯基用变换文法作为形式语言的描述手段。例如,L_a可用如下的变换文法描述:$\{S \to a, S \to aS\}$。这个文法由两条变换规则组成。每一步变换(也叫推导)都用一条变换规则的右部替换它的左部。例如,句子$aaaaa$可以这样推导出来:$S \to aS \to aaS \to aaaS \to aaaaS \to aaaaa$。

严格地说,变换文法定义成四元组$G = (\Sigma, V, S, P)$。Σ是字母表,又称终结符号表。V是变量表,又称非终结符号表;S是出发符号;P是变换规则(又称产生式)的集合,其中Σ, V和P都是有限集,$\Sigma \cap V = \varnothing, S \in V$。又令$\alpha$和$\beta$分别表示$(\Sigma \cup V)^+ - \Sigma^+$和$(\Sigma \cup V)^*$中的元素(用+代替*表示不含空字),则$P$中所有产生式皆形如$\alpha \to \beta$。这样定义的文法称为0型文法,又称短语结构文法,所生成的语言称0型语言。每个0型语言都是递归可枚举集(参见**图灵机**),反之亦然。

以$|\alpha|$表示符号串α的长度。对0型文法加限制$|\alpha| \leqslant |\beta|$,即得到1型文法,生成的语言称1型语言。1型文法也可以这样定义:它的所有产生式均取$\gamma A \delta \to \gamma \omega \delta$的形式,其中$\gamma, \omega, \delta \in (V \cup \Sigma)^*$, $|\omega| > 0, A \in V$。其直观意义是:在左有γ、右有δ的环境下,A可以被ω替换。因此,1型文法和1型语言又分别叫上下文有关文法和上下文有关语言。

如果要求0型文法中$\alpha \in V$,便得到2型文法,又称上下文无关文法,生成的语言称2型语言或上下文无关语言。不含空字ε的2型语言必可由不含空产生式$A \to \varepsilon$的2型文法生成,其中$A \in V$, ε在书写时可省去。

若要求2型文法中产生式的右端为aB或a,其中$a \in \Sigma, B \in V$,则得到3型文法,或称正则文法,又称右线性文法,生成的语言称为3型语言,或正则语言,或右线性语言。

以G_0, G_1, G_2, G_3分别代表上述四类文法,以L_0, L_1, L_2, L_3分别代表四类语言(其中L_2不含空字),又以L_r表示由递归集构成的语言类,则有$L_3 \subset L_2 \subset L_1 \subset L_r \subset L_0$,这些包含都是真包含。例如,取$\Sigma = \{a, b, c\}$, $L_0 = \left\{ \bigcup_{i=0}^{\infty} x_i \mid x_i \text{ 为任意一个由第 } i \text{ 台图灵机接受的语句} \right\}$是零型语言而不是递归集。$L_r = \left\{ \bigcup_{i=0}^{\infty} x_i \mid x_i \text{ 为任意一个不能由第 } i \text{ 个 1 型文法产生的语句} \right\}$是递归集而不是1型语言。$L_1 = \{a^n b^n c^n \mid n \geqslant 1\}$是1型而非2型语言。$L_2 = \{a^n b^n \mid n \geqslant 1\}$是2型而非3型语言。$L_3 = \{a^n \mid n \geqslant 1\}$是3型语言,这里$a^n$表示$n$个$a$的连接。

形式语言和自动机 上述文法和语言的分层方法,是乔姆斯基于1959年提出来的,因而称为乔姆斯基分层。这种分层法提出不久,人们即发现它和自动机的分类有密切的关系。到1964年,四类文法及其语言全部在自动机中找到了它们所对应的位

置。0 型、1 型、2 型和 3 型语言正好分别是**图灵机**、非确定型**线性有界自动机**、非确定型**下推自动机**和**有限自动机**接受的语言。确定型和非确定型图灵机接受的语言是相同的,确定型和非确定型有限自动机接受的语言也是相同的。确定型下推自动机接受的语言集合是三型语言的真子集,称为确定型上下文无关语言。至于非确定型线性有界自动机和确定型线性有界自动机接受的语言集合是否相同,是一个著名的尚未解决的问题,简称 **LBA 问题**。

形式语言的代数性质 研究形式语言的第三种途径是把它作为一种代数结构来考察。可以施行于语言的代数运算,包括求交($L_1 \cap L_2$)、求并($L_1 \cup L_2$)、求差($L_1 - L_2$),求补($\sim L$,即 $\Sigma^* - L$)、反演(L^{-1},$ab \in L \leftrightarrow ba \in L^{-1}$)、乘积($L_1 \cdot L_2$,$W_1 \in L_1$,$W_2 \in L_2 \to W_1W_2 \in L_1 \cdot L_2$)、乘幂闭包($L^{\circledast}$,$W \in L \to W^n \in L^{\circledast}$,$n \geqslant 0$)、无空字乘幂闭包($L^{\circledast}$减去空字 ε)、同态($L_1 \to L_2$)、无空字同态、置换($f(L)$,L 中每个字母置换成一个语言、字母串置换成与串中各字母对应之语言的乘积)、正则置换(置换语言都是正则语言)、L_1 对 L_2 左商($L_2 \backslash L_1 = \{Q \mid PQ \in L_1, P \in L_2\}$)、右商($L_1 / L_2$)等。早在 1961 年即有人指出:一个上下文无关语言和一个正则语言之交仍是上下文无关语言。不久人们发现,这一结果具有相当的普遍性:几乎所有重要的语言族都具有在与正则语言相交下封闭的性质。从 60 年代中期开始,研究某些语言族在某些代数运算下的封闭性已经成为形式语言代数研究的一个中心课题。乔姆斯基分层中各型语言在一些代数运算下的封闭性如表 1 所示。

表 1 四类语言在代数运算下的封闭性

代数运算 \ 语言族	L_0	L_1	L_2	L_3
求并	✓	✓	✓	✓
乘积	✓	✓	✓	✓
乘幂闭包	✓	✓	✓	✓
无空字乘幂闭包	✓	✓	✓	✓
求补	×	?	×	✓
求交	✓	✓	×	✓
与正则语言相交	✓	✓	✓	✓
反演	✓	✓	✓	✓
置换	✓	×	✓	✓
无空字置换	✓	✓	✓	✓
同态	✓	×	✓	✓
无空字同态	✓	✓	✓	✓
对正则语言的左商	✓	×	✓	✓
对正则语言的右商	✓	×	✓	✓
正则置换	✓	×	✓	✓
无空字正则置换	✓	✓	✓	✓

抽象语言族 简称 AFL,由 S. Ginsburg 等人于 1966 年提出,是研究语言族在代数运算下封闭性质的有力工具。令 Σ_∞ 为无限字母表,在其任一有限子集 Σ_i 上构造语言 $L_i \subseteq \Sigma_i^*$。如果任何一组语言 $\{L_i\}$ 中至少包含一个 $L_{i_0} \neq \varnothing$,则称 $\{L_i\}$ 为一语言族。在同态、逆同态和正则语言相交下保持封闭的语言族称为满三重组。对并运算封闭的满三重组称为满半 AFL。对乘幂闭包封闭的满半 AFL 称为满 AFL。从一个语言族 $\mathscr{L}$ 出发,经上述代数运算后得到的闭包分别称为由 $\mathscr{L}$ 生成的满三重组、满半 AFL 和满 AFL,以 $\hat{\mathscr{M}}(\mathscr{L})$、$\hat{\mathscr{G}}(\mathscr{L})$ 和 $\hat{\mathscr{F}}(\mathscr{L})$ 表之。如果语言族 $\mathscr{L}$ 只包含一个语言 L,则由 $\mathscr{L}$ 生成的结构分别称为满主三重组、满主半 AFL 及满主 AFL。如果把同态限制为无空字同态,即不得把非空字映为空字,则所有以上定义中的"满"字皆应除去。

一个语言族成为 AFL 的充分必要条件是它在并运算、无空字乘幂闭包、无空字正则置换、与正则语言相交及有界同态(对固定的 $k \geqslant 1$,任一语句的同态映象之长度不超过原长 k 倍)下是封闭的。一个语言族成为满 AFL 的充分必要条件是它在并运算、乘幂闭包、正则置换、与正则语言相交及同态映射下是封闭的。把乔姆斯基的四类语言看作语言族 $\mathscr{L}_0, \mathscr{L}_1, \mathscr{L}_2, \mathscr{L}_3$,则它们都是 AFL,其中 $\mathscr{L}_0, \mathscr{L}_2, \mathscr{L}_3$ 是满 AFL,$\mathscr{L}_1$ 不是,因为它在一般的同态映射下不封闭。

判定问题 一些已有的成果如表 2。表中 D 表示可判定,U 表示不可判定,G 表示文法,L 表示语言。

表 2 与形式语言有关的判定问题

判定问题 \ 语言族	L_0	L_1	L_2	L_3
任意字 $x \in L(G)$?	U	D	D	D
$L(G_1) \subset L(G_2)$?	U	U	U	D
$L(G_1) = L(G_2)$?	U	U	U	D
$L(G) = \varnothing$?	U	U	D	D
$L(G)$为无限集合?	U	U	D	D
$L(G) = \Sigma^*$?	U	U	U	D

续表

判定问题 \ 语言族	L_0	L_1	L_2	L_3
$L(G)\in L_3$?	U	U	U	/
$L(G)\in L_2$?	U	U	/	/
G 无二义?	/	/	U	D
$L(G)$ 无二义?	/	/	U	/
$\sim L(G)=\varnothing$?	U	U	U	D
$\sim L(G)$ 为无限集合?	U	U	U	D
$\sim L(G)\in L_3$?	U	U	U	/
$\sim L(G)\in L_2$?	U	U	U	/
$\sim L(G)$ 和 $L(G)$ 同一类型?	U	?	U	/
$L(G_1)\cap L(G_2)=\varnothing$?	U	U	U	D
$L(G_1)\cap L(G_2)$ 为无限集合?	U	U	U	D
$L(G_1)\cap L(G_2)\in L_3$?	U	U	U	/
$L(G_1)\cap L(G_2)\in L_2$?	U	U	U	/

非乔姆斯基的文法和语言类 基本原理是对每一步变换时允许使用的产生式加以不同的限制。例如,矩阵文法把产生式分成有限多个有序组,要求每一步变换必须连续地使用整组产生式。时控文法也把产生式分组,规定每一时刻 t 只能从第 t 组中选一产生式加以应用。以 $\varphi(t)$ 表示第 t 组产生式,则当有 t_0 使 $\varphi(t+t_0)=\varphi(t)$ 时,它称为周期时控文法。有序文法把产生式排成偏序,只有当排在前面的产生式不能应用时,才允许使用后面的产生式。Lindenmeyer 于 1968 年提出的 L 系统以**细胞自动机**和生物发育模型为背景。它规定在每一步变换中,对同样的左部符号必须使用同一个产生式。例如若 $B\to\alpha$ 和 $B\to\beta$ 同为产生式,则从 BAB 只能推出 $\alpha A\alpha$ 和 $\beta A\beta$,而不能推出 $\alpha A\beta$ 或 $\beta A\alpha$。L 系统的出发符号是一个字母串,一般没有终结符或非终结符之分。在名称上,它用 0 表示零面(产生式上下文无关),I 表示上下文有关,D 表示决定型,P 表示增值型(不产生空字),T 表示产生式按表格分组。例如,$DT0L$ 语言表示由决定型、表格型、零面 L 系统生成的语言。

ω 语言 令 $\Sigma^\omega=\left\{\prod_{i=1}^{\infty}a_i \,\middle|\, a_i\in\Sigma\right\}$,其中 Σ 为有限字母表,则 Σ^ω 的元素称为 ω 字,Σ^ω 的子集称为 Σ 上的 ω 语言。设 A 是以 Σ 为输入字母表的有限自动机,则 $(A,F)=A^\omega$ 称为 ω 有限自动机(ωFSA),其中 $F\subseteq 2^K$,K 是 A 的状态集。令 $\prod_{i=1}^{\infty}S_i$ 为 A^ω 在识别 ω 字 α 过程中经历的状态序列,则

1. $A^\omega 1$ 型接受 α,若 $(\exists H\in F)(\exists i)S_i\in H$;
2. $A^\omega 1'$ 型接受 α,若 $(\exists H\in F)(\forall i)S_i\in H$;
3. $A^\omega 2$ 型接受 α,若 $(\exists H\in F)(\forall k)(\exists i)S_i^k\in H$;
4. $A^\omega 2'$ 型接受 α,若 $(\exists H\in F)(\forall k)(\forall i)S_i^k\in H$;
5. $A^\omega 3$ 型接受 α,若 $\{S_i^k \mid i,k$ 任意$\}\in F$。

其中 S_i^k 是第 k 种在 $\prod_{i=1}^{\infty}S_i$ 中无穷次出现的状态,$S_i^k=S_i$。

麦克纳登证明了 ωFSA 的 2 型和 3 型接受以及 $\omega DFSA$ 的 3 型接受都是等价的(D 表示确定型),并以此定义 ω 正则语言。ω 正则语言在所有布尔运算下封闭。对任意 ω 正则语言,下列问题可判定:① L_1 为空、有限或无限,② $L_1=L_2$,③ $L_1\subseteq L_2$,④ $L_1\cap L_2=\varnothing$。

科恩等人用类似方法定义 ω 下推自动机 ωPDA,还定义了 $\omega 3$ 型文法 ωRG 和 $\omega 2$ 型文法 ωCFG,证明了 ωRG 和 ωCFG 生成的语言正好被 ωFSA 和 ωPDA 接受。

高维文法 上面讨论的文法只许 Σ 含简单字母,称为串文法。若允许 Σ 为树或图的集合,即是树文法和图文法,生成的语言称为树语言和图语言。这些文法统称高维文法,也有人研究更复杂的高维文法。

参考文献

1. Revesz G E. Introduction to formal languages. New York:McGraw-Hill,1983
2. Book R V. Formal language theory perspectives and open problems. New York:Academic Press,1980

(陆汝钤)

xingshi yuyi

形式语义(formal semantics) 用数学方法,尤其是形式系统严格定义出的语言的语义。程序设计语言是人们用来和计算机系统进行通信和控制其工作的人工语言。作为语言,人工语言和自然语言(如汉语、英语等)一样,有其语法、语义和语用范畴。程序设计语言的语法是指程序的组成规则,语义是指程序的含义。

程序设计语言的语义通常是由设计者用一种自然语言非形式地解释的,实现者和使用者则依据各自的理解去实现和使用这种语言。然而使用自然语言和非形式的方法解释语义,容易产生歧义,造成语言设计者、用户和实现者对语义的不同理解,影响语

言的正确实施和有效使用。程序设计语言中的过程调用语句就是这方面的一个典型例子。人们发现对过程调用语句的非形式解释可能导致各种不同的理解,产生多种不同的效果。

为了正确、有效地使用程序设计语言,必须了解语言中各个成分的含义,并且要求计算机系统执行这些成分所产生的效果与其含义完全一致。人们这种对语义精确解释的要求便产生了形式语义学。形式语义学的研究始于20世纪60年代初期,在程序设计语言ALGOL 60的设计中,第一次明确区分了语言的语法和语义,并使用BNF符号系统成功地实现了语法的形式描述。语法的形式化大大推动了语义形式化的研究,围绕ALGOL 60的语义出现了形式语义学早期的研究热潮。以后的程序设计语言,如PASCAL,Ada等,都有人给出了严格的形式语义,旨在为编制程序语言的编译程序提供正确依据。

美国斯坦福大学J. McCarthy于1962年系统地论述了程序设计语言语义形式化的重要性,以及它同程序的正确性、语言的正确实现等的关系,并提出在形式语义研究中使用抽象语法和状态向量等方法。近年来,形式语义的理论和应用都有了很大发展。

程序设计语言的语法是规定程序组成方法的一些规则,称为具体语法,但在定义程序的语义时,必须首先识别给定的程序,分析程序的语法结构。因此,在形式语义中使用一种讨论程序分解的语法规则,这种语法称作抽象语法。不同的程序设计语言往往使用不同的记号和表示方式。形式语义提供的方法适用于一切程序设计语言,故抽象语法采用的记号和表示方式也是具体语法的一种抽象。

在定义程序设计语言的语义时,需要一种定义语义的语言,这种语言称为元语言。元语言可以采用已有的数学语言,也可以是以数学理论为基础的专门设计的语言。用元语言去定义程序语言的形式语义,必须首先严格定义元语言的语义。

用程序设计语言编写的程序,规定了它对计算机系统中数据的一个加工过程,形式语义的基本方法是将程序加工数据的过程及其结果形式化,从而定义程序的语义。

由于形式化中侧重面和使用的数学工具不同,形式语义可分为四大类。①**操作语义**:着重模拟数据加工过程中计算机系统的操作;②**指称语义**:主要刻画数据加工的结果,而不是加工过程的细节;③**公理语义**:用公理化的方法描述程序对数据的加工;④**代数语义**:把程序设计语言看作是刻画数据和加工数据的一种抽象数据类型,使用研究抽象数据类型的代数方法,来描述程序设计语言的形式语义。

参考文献

周巢尘.形式语义学引论.长沙:湖南科技出版社,1985

(周巢尘 李晓山)

xuni cidaiku

虚拟磁带库(virtual magnetic tape library, VTL) 一种用硬磁盘来仿真**磁带库**的存储备份设备。这种技术在设备内部实际使用硬磁盘组进行数据存储,通过虚拟化技术将硬磁盘空间虚拟化成磁带以及磁带库,从而对现有备份软件呈现出与传统物理磁带库一样的访问接口和逻辑视图。由于硬磁盘与磁带的性能差异,采用磁盘存储的虚拟磁带库的性能远远高于传统的物理磁带库。

传统的物理磁带库是基于磁带的备份系统,由多个**磁带机**、多个槽、机械手臂组成,并可由机械手臂自动实现磁带的拆卸和装填,支持多个磁带机并行工作,能够提供自动备份和数据恢复功能,并可在管理软件的支持下实现智能恢复、实时监控和统计,是集中式网络数据备份的主要设备。但其内部机械手以及磁带机属于精密机械设备,故障率高,速度慢,数据备份恢复时间比较长,且磁带介质易受粉尘、湿度、粘连、霉点等因素的影响,对保存环境的要求比较高。鉴于传统磁带库的这些缺陷,IBM公司在1997年就提出了第一个虚拟磁带库IBM虚拟磁带服务器(VTS),随后磁带机厂商StorageTek(2005年被SUN公司收购,接着被Oracle收购)也推出了类似存储解决方案VSM,直接冲击了在备份领域处于垄断地位的磁带库市场。

随着磁盘技术的快速发展,单位容量磁盘存储价格急剧下降,采用磁盘进行存储的虚拟磁带库数据保护成本已大大降低。近年来虚拟磁带库产品广泛采用廉价的并行ATA(PATA)(参见**外存储设备接口**)或串行ATA(SATA)接口**磁盘阵列**进行内部数据存储,利用**磁盘阵列**技术一方面可有效提升系统性能与可靠性,另外也可以方便地利用额外的阵列机箱进行系统扩容,从而提升系统可扩展性。通过将数据备份到快速的磁盘而不是慢速的磁带,磁带库数据备份以及恢复性能大幅提升。虚拟磁带库不用额外的硬件开支即可允许用户灵活配置虚拟磁

带驱动器个数、虚拟磁带盒和指定磁带盒容量。由于以上诸多优势，将数据备份到虚拟磁带库的方法正在得到大型数据中心的认可，越来越多的用户开始淘汰传统物理磁带库，而采用虚拟磁带库作为主要的备份设备。但对于需要长期归档的数据，传统的物理磁带库在保存数据的可靠性和低能耗方面仍具有虚拟磁带库难以替代的优势。

目前市场上的虚拟磁带库依照架构不同可分为三种类型：磁盘阵列型、应用服务器型、备份软件型。

（1）磁盘阵列型　它是以磁盘阵列为基础发展的虚拟磁带库，通过内置于磁盘阵列控制器内的虚拟软件，将磁盘阵列存储空间仿真成磁带库具备的所有特征。这种类型的虚拟磁带库用户以大型企业为主，采用光纤信道主机接口，支持光纤信道存储区域网（FC SAN）网络环境。该方案突破了操作系统和 PC 服务器架构的限制，使虚拟磁带库真正成为了一种独立的外设，其使用方式也更接近普通磁带库，其优越性能也体现得更加充分。

（2）应用服务器型　将虚拟磁带库管理软件安装在一台独立的专用服务器内，将该服务器及所连接的磁盘存储设备模拟成磁带库，与备份服务器端可以借由 SCSI、iSCSI 接口或光纤信道等与传输接口相连，该方案部署应用上较有弹性，速度比较快，且数据受主机的影响小，不足是需要专门利用一台服务器作为虚拟磁带库管理器，系统优化性略低。

（3）备份软件型　直接将虚拟磁带库功能整合至备份软件内，将备份服务器的文件系统分区模拟成磁带库，从而使备份软件以磁带库方式使用磁盘文件系统。这种方式成本比较低，但备份主机的负担较大，且无法和其他厂牌备份软件搭配使用。

（谭志虎）

xuni cunchuqi

虚拟存储器（virtual memory）　在具有层次结构存储器（参见**分层存储器体系结构**）的计算机中，为用户提供比**主存储器**容量大且可随机访问的地址空间。虚拟存储器的地址称为**虚地址**或逻辑地址。程序运行时，**中央处理器**实际访问的存储器仍然是主存储器，主存和辅存（**辅助存储器**）称为实际存储器，简称实存。实存的地址称为**实地址**或物理地址。

虚拟存储器的概念于 1961 年由英国曼彻斯特大学提出，并在 Atlas 计算机中实现。Atlas 计算机的主存储器容量为 16 KB，外存储器（磁鼓存储器）的容量为 96 KB。在现代计算机中，虚拟存储器已得到广泛应用，不但用于**大型计算机**，也用于**小型计算机**和**微型计算机**。

虚拟存储器的虚实地址转换，块映射方式和块替换策略在原理上和高速缓冲存储器的类似。它们的主要不同之处是**高速缓冲存储器**的机制基本上由硬件实现，而虚拟存储器的机制由**操作系统**在硬件的配合下实现。硬件主要负责虚实地址转换，操作系统负责调页管理、实存管理、主存和辅存之间的信息调度等。

对于具有虚拟存储器的计算机，编译程序产生的程序是用虚地址编程的，程序由操作系统装入辅存中。程序运行时，虚地址被转换为实地址，如果所需的内容已在主存中，则中央处理器访问主存的实地址；如果所需的内容不在主存，称为页面失效，计算机产生页面失效中断，操作系统将要访问的一页从辅存中调入主存。

虚实地址转换

（1）页式虚拟存储器的虚、实地址转换在页式虚拟存储器中完成。以页为基本单位的虚拟存储器叫作页式虚拟存储器。主存空间和虚存空间都划分成若干个大小相等的页，主存（即实存）的页称为实页，虚存的页称为虚页。主存的页面按顺序编号，称为实页号。**程序**也划分为大小与页面相同的页，并按顺序编号，称为虚页号。程序中的页可以装在主存的不连续的任意页面中。虚页向实页的映射（即虚页号和实页号的对应关系）由页表给出。每一个装入计算机的程序（进程）都有自己的页表。图 1 为页表的简单例子。设在主存中已装入 A、B、C 3 个进程，现在要调入辅存中的占有 3 个页面的进程 D。操作系统将进程 D 的 3 个页装入主存中的 3 个不连续的空闲的页面，并建立进程 D 的页表。页表中的一行称为页表项。每一页表项有若干个控制位，其中 1 个位为装入位，它为 1 时表示该页已在主存中，为 0 时表示不在。根据页表可将虚地址转换成为实地址。

页表是操作系统根据主存的运行情况自动建立的，对程序员透明。页表平时可以存放在辅存中，在需要时调入主存。在主存中有存放页表的页表区。每个页表都有自己的页表起始地址。在程序运行时，存储管理软件将该程序的页表起始地址送到页表基址寄存器，如图 2 所示，页表起始地址和虚地址中的虚页号拼接成页表地址，从而可在页表中找到实页号。如果页表项中的装入位为 1，表示该页已

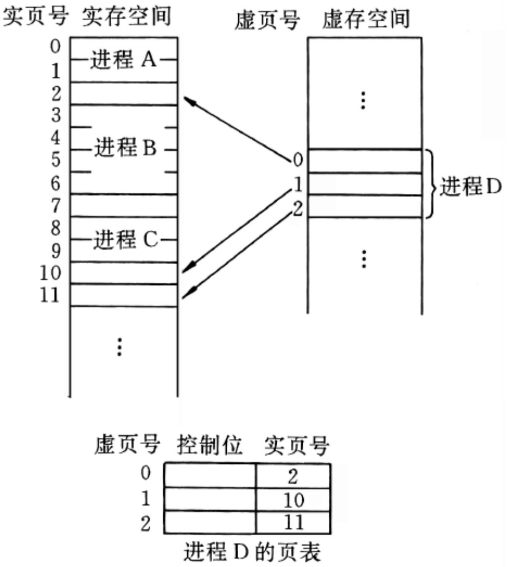


图 1　页表举例

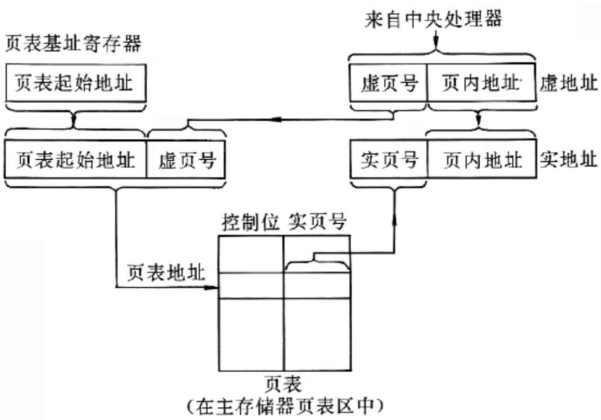


图 2　页式虚拟存储器的虚实地址转换

调入主存储器，页面有效，可根据此实页号去访问主存；如果装入位为 0，则表示该页未调入主存，产生缺页中断，操作系统将所需的页从辅存调入主存。

（2）段式虚拟存储器的虚实地址转换在段式虚拟存储器中完成。段是按照程序的逻辑结构划分的，段的长度因程序不同而不同。虚地址由段号和段内地址组成。为了将虚地址转换成实地址，需要一个段表。段表指明程序中的各个段在主存储器中的实际位置，凡装入计算机的每一个程序都有自己的段表。段表由段表项组成，每一段表项包含该段在主存中的起始地址、段的长度以及装入位等控制位。段表本身也是一个段，平时可存放在辅存中，需要时调入主存。和页式虚拟存储器相似，在访问存储器时，要先查阅段表，在段表中去寻找实地址。

（3）段页式虚拟存储器的虚实地址转换　在段页式虚拟存储器中完成。段页式虚拟存储器是段式虚拟存储器和页式虚拟存储器的结合。主存分为页面，程序分为段，每个段又分为若干个和主存页面同样大小的页。虚地址包含段号、页号、页内地址等，凡装入计算机的每一个程序都有 1 个段表和 1 组页表（每一段有 1 个页表）。段表项中有指明该段的页表在主存中的起始地址和页表长度等信息，每个段的页表则进一步指明该段中的各页在主存中的实际位置。段的起始地址不能是任意的，必须位于主存页面的起始单元中。每次访问存储器时，要先查段表，得到页表的起始地址后，再查页表，才能找到实地址。

在上述 3 种虚实地址转换时，由于页表和段表都存放在主存中，需要事先访问主存一两次，从而使中央处理器访问存储器的时间加长。为了缓解这个问题，在虚拟存储器中，可增设**转换检测缓冲器**，它使虚实地址的转换加快。

（4）虚地址到辅存实地址的转换　在缺页中断时，操作系统应将所需的虚页从辅存调入主存，调页时虚地址应转换成辅存实地址。现代计算机一般都采用**磁盘存储器**作为辅助存储器，磁盘的盘片表面上的磁道划分为若干个扇区，每个扇区的大小通常等于 1 个主存页面。磁盘的实地址由磁盘机号、柱面号、磁头号和扇区号组成，虚页号与磁盘实地址的对应表称为外页表，外页表也是按虚页号的顺序排列的，外页表项的格式示于图 3。外页表的内容是在将程序装入辅存时填写的，其中 M 是装入位，M 为 1 时，表示该扇区中的页已从海量存储器装入磁盘。通常外页表存放于辅存中，当发生缺页中断需要查用时才将它调入主存。从虚页号找外页表可由软件实现。

M	磁盘机号	柱面号	磁头号	扇区号

图 3　外页表项的格式

虚拟存储器工作过程

虚拟存储器工作过程可归纳如下（以页式虚拟存储器为例）：

（1）根据虚地址中的虚页号查页表。

（2）当页表中对应于该虚页的页表项中的装入位为 1 时，表示该页已在主存中，页表项中的实页号和页内地址拼接为主存实地址。

（3）按主存实地址去访问主存（绝大部分主存访问到此为止）。

（4）当页表中对应于该虚页的页表项中的装入位为0时，产生缺页中断。

下面转入管态，由操作系统处理缺页中断：

（5）根据虚地址查外页表。

（6）当外页表中对应于该虚页的外页表项中的装入位为1时，表示该页在辅存中，从外页表项可得到辅存实地址，按辅存实地址去访问辅存中的这一页。

（7）查主存页面表，以确定从辅存中调出的页放入主存中的位置。

（8）如果主存未装满，只需确定1个空的页面即可。

（9）如果主存已装满，根据替换算法确定替换页面。

（10）将上述（8）和（9）确定的空页面或替换页面的实存页面号送入通道（或输入输出处理机）。

（11）在进行页面替换前检查被替换的页在进入主存之后是否已被修改，如果未被修改，通道从辅存中读出要调出的页并写入主存的指定页面。

（12）如果被替换的页在主存中已被修改，需先将该页写回它在辅存中原来的位置，然后再从辅存中读出要调出的页并写入主存的指定页面。

缺页中断的处理到此结束，下面转回目态，原来被中断的程序继续运行。

（13）如果在执行完上述（5）以后，在外页表中查出对应于该虚页的外页表项中的装入位为0，则表示该页不在辅存中，这时需要进入另一层中断，将海量存储器中的这一页调入辅存，然后退出这层中断，继续执行（6），将这一页从辅存调入主存。

参考文献

孙强南，孙昱东．计算机系统结构．北京：科学出版社，1992

（孙强南　张广艳）

xunihua

虚拟化（virtualization）　将实体计算资源进行逻辑抽象而创建虚拟计算资源的方法与过程。由虚拟化创建的虚拟资源通常比实体资源具有更丰富的功能、更灵活的可配置性或者更友善的应用接口。虚拟化的主要作用便是使资源更适于应用。

逻辑抽象和虚拟化是操作系统的两个十分重要且紧密相关的基本机理。

整体上，操作系统将硬件裸机改造成为一台资源利用率更高，运行环境更好，使用和管理更加方便、功能更加强大的虚拟机器。

在资源管理方面，为提高物理资源的利用率以及达到多用户共享一套计算机物理资源的目的，操作系统将物理上的一个资源变成逻辑上的多个资源，给每个用户形成独占资源的假象，SPOOLing是典型的例子。操作系统通过空间分割共享（如存储器）或分时共享（如处理器）创建出许多虚拟资源，应用程序在分得的空间上或时间间隔内享有独占控制权。虚拟存储系统把内存（或称主存）和磁盘统一管理起来，用内存作为磁盘的高速缓存，以此操作系统为用户提供了远比物理内存大得多的虚拟内存空间。为解决外部设备的易用性问题，对应用屏蔽硬件资源的物理特性和接口细节，操作系统为各种设备配置了设备驱动程序。文件可以视为磁盘等设备的逻辑抽象和虚拟化，操作系统将文件中的字节映射到设备物理块，使程序员只需通过文件调用来控制文件，而无须关心文件在哪个磁盘的哪个具体位置以及磁盘输入输出的所有细节。

在程序控制方面，进程是对进入内存正在运行的程序在处理器上操作的状态集的一种逻辑抽象，一个程序可以创建出多个进程，因此，进程是程序虚拟化的产物。进程的创建隐含着对处理器与内存资源的双重需求，每个进程拥有独立的虚拟内存空间，给每个进程造成独占整个内存的假象。管程是基于抽象数据类型原理和虚拟化技术提出的一种同步机制，用于协调各种进程间的同步互斥问题。

在计算机科学中，操作系统能够将很多台计算机组成的集群系统创建成一台具有单系统映象的虚拟机器，这是多对一的虚拟化，主要是为了提高整个系统的可扩展性、可用性和资源利用率；也能将一台计算机变成几台甚至上百台相互隔离的、运行不同操作系统实例的虚拟计算机，以实现不同应用在虚拟化平台上整合，简化系统的部署和管理，这是一对多的虚拟化，主要是为了提高资源的利用率以及让信息技术对未来业务的变化更具适应力。

目前，操作系统呈现出多平台统一的发展趋势，同一个操作系统通过构件技术可以虚拟地进行灵活扩展和变化，既支持桌面系统，也支持嵌入式系统，甚至支持数据中心。随着云计算和现代软件服务业的发展，基于虚拟机的主机租赁服务的市场需求越来越大，因此，一对多的服务器虚拟化是目前虚拟化研究的热点。服务器提供的是一个完整的系统平台，运行的是一个完整的操作系统，称为宿主操作系统；而虚拟机上可以通过对多个客户机或目标环境的模拟仿真，运行多个客户操作系统。这些客户操

作系统实际上是建立在一个宿主操作系统之上通过虚拟机监视器管理的。虚拟机监视器是一种中间层软件，是系统虚拟化技术的核心。因此，虚拟机技术使多个客户操作系统可以同时运行在同一计算机上，它们之间是相互隔离的，且虚拟机可以提供不同于宿主计算机的指令集，容易做到应用开通、维护、高可用性和灾难恢复。当然，虚拟机在访问硬件资源时会比物理机低效。当多个虚拟机同时运行在同一物理机上时，为了避免每个虚拟机因负载的变化而可能出现的执行速度等性能不稳定问题，需要完整的虚拟机实例之间的资源隔离机制。

虚拟化已成为云计算的核心技术。多对一的虚拟化属资源聚合，一对多的虚拟化属按需服务，两者是密切关联的。上述虚拟化过程应在硬件、网络、存储、数据、软件和平台各个层次展开。为了将云计算的各类实体资源变换成同构的虚拟资源，需要构建一个完整的信息资源的虚拟化环境。如何通过多层次多方位的虚拟化技术，实现云计算平台各类信息资源的无差别共享，使云计算平台成为集中式同构无限可扩展的网络计算平台，是云计算领域需要不断深入探索的课题。

参考文献

1. http://en. wikipedia. org/wiki/Virtualization

2. http://en. wikipedia. org /wiki /Virtual_machine

（章文嵩　吴泉源）

xunihua jishu

虚拟化技术(virtualization technology)　实现计算机系统中上、下层或软、硬件层次间相耦合的一种技术。虚拟化是一个广义的术语，通常指计算、存储、网络等计算机的硬件资源或进程、线程乃至操作系统以虚拟方式而非在真实基础上运行。虚拟化技术可以虚拟方式扩大硬件的容量，简化软件的重新配置过程，甚至可以实现在同一个硬件平台上支持多个操作系统的同时运行，从而显著提高计算机的工作效率。

发展简史

虚拟化技术的概念在现代操作系统的设计与实现中已有所体现，例如 32 位操作系统的虚拟内存技术，采用内存页面与外存（主要是硬盘）间换入换出的办法，为系统的每个执行实体（进程）虚拟出 4 GB 内存空间（尽管计算机实际物理内存远远小于 4 GB）。同时，在现代多任务操作系统中，每个执行实体在运行过程中都会感觉自身独占处理器资源，而实际上，它们采用分时复用的思想，由操作系统调度，多个进程轮转执行，这也是虚拟化思想体现在现代操作系统中的一个例子。

在虚拟化技术萌芽的 20 世纪 60—70 年代，人们研究虚拟化技术的目标是对相对昂贵的硬件资源进行充分利用，通过虚拟化手段让更多的人能够通过终端设备接触和使用计算机系统。但是，到了 20 世纪七八十年代，随着大规模集成电路的出现和个人计算机的普及，计算机硬件变得越来越便宜。当初为共享昂贵硬件而设计的虚拟化技术慢慢无人问津，而只是在高档服务器（如 IBM 小型机）中继续存在。近年来，虚拟化技术重新受到关注，其原因主要有以下两个方面：其一，计算机系统经过多年发展，在变得越来越强大的同时，也在变得越来越难以管理（例如今天的网络系统、分布式计算系统），软硬件管理开销（特别是电费开销）也逐年增加。特别是随着处理器多核化（参见**多核处理器**）时代的到来及冗余计算资源的引入，这一矛盾势必越来越尖锐。其二，今天的计算，已经从以前以计算机为中心向以用户为中心的服务计算过渡，人们更关心的是计算系统能够为用户提供怎样的接口和提供怎样的服务，以适应用户复杂和多样化的需求。在这一背景下，由于虚拟化技术既能够屏蔽底层复杂的物理环境，又能够为用户提供可配置的使用环境，就重新受到工业界和学术界的关注。

20 世纪 90 年代末期，VMware 和其他虚拟化软件厂商率先为虚拟化技术在 x86 服务器环境下开辟了道路，使得虚拟化应用的前景更加广阔。他们实施的是一种软件解决方案，以虚拟机监视器（VMM）为中心使 PC 服务器实现虚拟化。然而，在这种纯软件的“完全虚拟化”模式中，每个客户机操作系统获得的关键平台资源都需要 VMM 控制，并由它来分配，以避免发生冲突。为了处理这些相关的控制，需要利用二进制转换来进行操作，而二进制转换的开销使得“完全虚拟化”的性能大打折扣。为解决这个问题，人们提出了“半虚拟化”技术。半虚拟化的实施不需要二进制转换，而是通过对客户机操作系统进行代码级修改，从而为操作系统提供新的接口，以使新的、定制的客户操作系统获得额外的性能和高扩展性。但是，修改客户机操作系统非常烦琐，带来了一些系统指令级别冲突以及运行效率的问题，需要软件开发商、集成商等投入大量的时间和人力对系统进行优化。此时，虚拟化技术的发展走到了硬件支持阶段，使半虚拟化的障碍得到了解决。

硬件虚拟化技术就是把纯软件虚拟化技术的各项功能用硬件电路来逐一实现。作为发挥多核处理器性能的一个有效手段，芯片制造厂商英特尔和超威半导体公司都在硬件级提供了对虚拟化的支持。在消除了对 CPU 半虚拟化和二进制转换技术的需求后，硬件已经能支持多种未经修改的客户机操作系统直接运行，VMM 的设计得到极大简化，进而使 VMM 能够按通用标准进行编写，减少了相关的性能开销，性能更加强大。

回顾自 20 世纪 90 年代起这一阶段的虚拟化技术发展历程，从纯软件的虚拟化（完全虚拟化、半虚拟化），再到硬件支持的虚拟化，突破了在 x86 平台上进行虚拟化的瓶颈，进驻的应用领域飞速增加。逐步扩展的技术路线证实了虚拟化技术在不断的挑战和创新中逐渐完善与成熟。

分类和技术内容

广义的虚拟化技术可以分为网络级虚拟化技术和系统级虚拟化技术。

1. 网络级虚拟化主要在网络计算中间件以及应用层进行研究与开发。从 20 世纪 90 年代后期开始，以**网格计算**为代表的基于网络计算中间件的虚拟化技术得到快速发展，其目标是共享和整合广域分布的网络资源，为用户提供虚拟网络计算环境。在学术界，大量研究机构对网格计算中的资源管理、任务调度、信息服务、数据管理、安全服务、编程环境等领域开展了研究工作，形成了以 Globus 等为代表的一批网格中间件系统；在工业界，以 IBM、HP、SUN、Oracle 等为代表的大型企业也积极推出自己的网格计算方案和产品。在网格项目开发上，大型网格项目持续得到众多国家的支持，应用的范围和规模持续增大，为一系列科学应用提供基础支撑。随着云计算的普及，网络虚拟化还包括云资源管理平台等重要内容。

2. 系统级虚拟化则侧重于计算机系统结构和底层系统软件。依据在计算机体系结构抽象层次的不同，系统级虚拟化技术可以分为指令级虚拟化技术、硬件虚拟化技术、操作系统虚拟化技术、编程语言级虚拟化、程序库级虚拟化技术、桌面虚拟化技术等多个类别。

（1）指令级虚拟化技术　指令级虚拟化通过纯软件方法，模拟出与实际运行的应用程序（或操作系统）所不同的指令集去执行，采用这种方法构造的虚拟机（VM，virtual machine）一般称为模拟器（emulator）。一个典型的计算机系统由处理器、内存、BUS、硬盘驱动器、磁盘控制器、定时器、多种 I/O 设备等部件组成。模拟器通过将客户虚拟机（guest VM）发出的所有指令翻译成本地指令集，然后在真实的硬件上执行，这些指令包括典型的处理器指令和特殊的 I/O 指令。当然，一个模拟器要成功地模拟一个真实的机器，它必须能够模拟真实机器所能做的一切事情。因为是通过模拟完成一切计算机的指令，因此它会引入较大的性能损失。

（2）硬件虚拟化技术　硬件虚拟化技术可以将虚拟资源映射到物理资源并在虚拟机环境中使用本地硬件。当模拟机需要访问关键物理资源时，模拟器接管其物理资源并妥善地多路复用。这种虚拟化技术要能够正确工作，所构造的虚拟机（VM）必须对其中的一些特权指令（例如修改页表等操作）进行处理，执行时产生陷入并将它传递给下层虚拟机管理器（VMM）执行。这是因为在 VM 中运行的未加修改的操作系统会利用特权指令得到 CPU 和内存资源。当某特权指令执行时产生一个陷入，便马上将指令发送给 VMM，这使得 VMM 可以完全控制虚拟机并保持每个 VM 隔离。然后，该 VMM 在处理器中执行该指令，并将模拟结果及特权指令返回给 VM。大多数商业虚拟机软件，都使用像代码扫描和动态指令重写这样的技术来解决这些问题。

（3）操作系统级虚拟化技术　一个应用的操作环境包括操作系统、用户函数库、文件系统、环境设置等。如果应用系统所处的这些环境能够保持不变，那么，应用程序自身无法分辨出其所在的环境与真实环境之间的差别。操作系统级虚拟化技术的关键思想在于，操作系统之上的虚拟层按照每个虚拟机的要求为其生成一个运行在物理机器之上的操作系统副本，从而为每个虚拟机产生一个完好的操作环境，并且实现虚拟机及其物理机器的隔离。

（4）编程语言级虚拟化　随着 Java 虚拟机（JVM）的到来，使得这种新的实现虚拟机的方式逐渐引起人们的注意。这种抽象层次的虚拟化技术的主要思想是在应用层次上创建一个和其他类型虚拟机行为方式类似的虚拟机，并支持一种新的自定义的指令集（例如 JVM 中的 Java 字节码）。这种类型的虚拟机使得用户在运行应用程序的时候就像在真实的物理机器上一样，并且不会对系统的安全造成威胁。像普通的机器一样，它通过安装一个商业的操作系统或利用其自身的环境来为应用程序提供操作环境。这种抽象层次的虚拟化系统主要包括 Java 虚拟机、Microsoft.NET CLI、Parrot 等。

(5) 程序库级虚拟化技术　在几乎所有的系统中,应用程序的编写都使用由一组用户级库来调用的 API 函数集。这些用户级库的设计能够隐藏操作系统的相关底层细节,从而降低普通程序员的软件开发难度。这部分所谈论的例子都工作在操作系统层上,并且创造了一个与众不同的虚拟环境,在底层系统上实现了不同的应用程序二进制接口(ABI)和不同的应用程序编程接口(API)。这种技术能很好地完成 ABI/API 仿真工作。

(6) 桌面虚拟化技术　桌面虚拟化是近几年来逐步兴起的一种虚拟化技术,它可为终端用户提供与传统桌面无差别的用户体验去使用应用服务,但是,这些桌面服务其实都是运行在远端服务器集群上。为了增强用户桌面部署的灵活性和资源管理的智能性,通常,远端服务器集群都是基于系统级虚拟化技术的。

在多核体系结构的快速发展背景下,虚拟化技术将可以有效地提高其资源利用效率;而在云计算背景下,虚拟化技术可以为云服务提供弹性的基础设施。同时,未来虚拟化技术还需要在 I/O 虚拟化、显卡虚拟化的性能提升以及桌面虚拟化、嵌入式设备虚拟化等方面继续发展。

参考文献

1. Creasy R J. The origin of the VM/370 time-sharing system. IBM Journal of Research and Development. 1981, 25(5): 483-490

2. Jin H. ChinaGrid: Making Grid Computing a Reality. In: Proceedings of 7th International conference on Asian Digital Libraries. Lecture Notes in Computer Science, 2005, 3334: 13-24

3. 金海,等. 计算系统虚拟化: 原理与应用. 北京: 清华大学出版社,2008

(廖小飞)

xuni huanjing shengcheng

虚拟环境生成(virtual environment generation)　利用计算机构建一个虚拟的而又能让用户在视觉、听觉、触觉、嗅觉等方面具有逼真感受和体验的环境的方法和技术,目的是提高虚拟现实的真实感、**沉浸感**和交互性。**虚拟现实**所产生的虚拟环境是人工构造的。它可以是某一特定现实的真实体现,也可以是纯粹假想的虚拟世界。

虚拟环境生成包括三维虚拟视景生成、三维空间声音生成、力触觉生成以及场景变化与声音和触觉力觉等信息的同步,从而支持自然的人机界面,并使得用户有沉浸感。

虚拟视景生成是指在显示设备上显示虚拟世界的技术,主要包括建模和绘制两个方面。建模主要是建立虚拟世界的数字模型,而绘制是指使用**计算机图形学**技术实时生成虚拟世界的三维表现,实时绘制技术包括多细节层次模型、可见性裁剪、对象纹理和并行处理等。

虚拟声音生成是虚拟现实的听觉表现方式,它基于三维空间声模型为虚拟现实中的使用者提供合成的声音信号,用于提高虚拟现实的真实感、沉浸感和交互性。三维虚拟声音生成的实现分为 3 个阶段,即定义-建模-绘制,声音绘制类似于上面的图形绘制。

力触觉生成最早来源于希腊语 haptesthai,意思是接触或触摸,它是力觉和触觉的总称。力觉/触觉生成是借助一些机械和电子设备,能向人提供物理刺激如机械力、振动、位移等,从而模拟人与真实环境接触时的力觉与触觉感受。

场景变化与声音和触觉力觉等信息的同步是虚拟环境生成的一项重要内容,三维声音的强度会随着用户在场景中位置的改变而变化,而触觉、力觉的变化也通常是与场景的变化或用户的动作相关的。只有做到这些信息的同步,才能很好地维持虚拟现实技术中的沉浸感,让用户感觉到自己好像完全置身于虚拟世界之中,成为虚拟世界中的一部分。使用户借助于自然交互由被动的观察者变成主动的参与者,沉浸于虚拟世界之中,参与虚拟世界的各种活动。

随着场景复杂性的增加,虚拟环境生成面临的挑战包括真实感、交互自然性、实时性等方面。

(潘志庚)

xuniji

虚拟机(virtual machine)　带有解释器(或称解释程序)的抽象机。虚拟机通常用来实现强调移植性的**高级语言**。高级语言程序先编译为抽象机代码指令序列,然后由**汇编语言**或 **C 语言**等实现的解释器加以解释执行。

JAVA 虚拟机 JVM 是一个典型的例子。JVM 规范定义了一个包括指令系统、寄存器集、堆栈、无用单元堆和存储区的抽象机。JVM 的实现可以基于实际的处理机指令,也可以基于微芯片。一旦系统安装了 JVM,编译后的 JAVA 程序,即 JAVA 字节代码,便可在其上运行。JAVA 虚拟机使 JAVA 编写的

程序一次编译到处可用成为可能。

运行 JVM 字节代码的工作是由解释器完成的。字节代码的解释分为三步：代码装入、代码校验和代码执行。字节代码的执行有两种方式：即时编译方式，解释器先将字节码编译成机器码，然后再执行该机器码；解释执行方式，解释器通过每次解释并执行一小段代码来完成 Java 字节代码程序的所有功能。

常见的虚拟机还有并行虚拟机（PVM），虚拟 Lisp 机（VLM）等。

JAVA 虚拟机 JVM 是基于栈的，另外一些虚拟机，如 Perl 6 的虚拟机 Parrot 是基于寄存器的。而并行虚拟机 PVM 则表示一些由网络连接的处理器的集合，其中每个处理器既可以对局部存储器进行存取，也可以对网络上的共享存储器进行存取，它甚至不要求处理器所在的每台机器都运行相同的操作系统。

虚拟机的另外一种含义是物理计算环境的软件模拟，通过系统控制程序为不同的**操作系统**或**程序**的执行提供的一个模拟的执行环境，其目的是在单一的机器上支持不同操作系统或一个操作系统的不同版本的同时运行，这方面的例子有 VMWARE 和 IBM 的 VM/ESA 等。

有时，虚拟机也指多用户操作系统为用户提供的共享资源环境，其中每个用户的感觉是独享所有计算机资源。 （黄林鹏）

xuni juyuwang

虚拟局域网（virtual local area network, VLAN） 将一个**局域网**（local area network, LAN）根据工作组、应用等逻辑上划分成多个网段，每个 VLAN 是一个广播域，与用户的物理位置没有关系。VLAN 内的主机间通信就和在一个 LAN 内一样，而 VLAN 间则不能直接互通。

VLAN 的协议和标准

目前最广泛使用的 VLAN 协议标准是 IEEE 802.1Q，在源 MAC 地址和以太网络类型 Ethertype 域之间增加了 4 个字节。

VLAN 帧结构见图 1，其中：

TPID 为标签协议字段，值为 0x8100，为 802.1Q 标记帧。

TCI 为标签控制信息字段，包括用户优先级、规范格式字段和 VLAN ID。

PCP：定义用户优先级。

16 bits	3 bits	1 bit	12 bits
TPID	TCI		
	PCP	CFI	VID

图 1 VLAN 帧结构

CFI：规范格式字段，在以太网交换机中，CFI 总被设置为 0。

VID：VLAN 标识是对 VLAN 的 12 位识别字段，VID = 0 用于识别帧优先级，4095（FFF）作为预留值，所以 VLAN 配置的最大可能值为 4094。

VLAN 的优点

（1）减少广播风暴。VLAN 能将网络划分为多个广播域，减少参与广播风暴的设备数量，从而有效地控制广播风暴的发生，防止广播风暴波及整个网络。

（2）增强网络安全。VLAN 可以将含有敏感数据的用户组与网络的其余部分隔离，从而降低泄露机密信息的可能性。不同 VLAN 内的报文在传输时是相互隔离的，即一个 VLAN 内的用户不能和其他 VLAN 内的用户直接通信，如果不同 VLAN 要进行通信，则需要通过路由器或三层交换机等三层设备。

（3）提高通信性能。VLAN 可以减少网络上不必要的流量，节省了带宽，从而提高了网络处理能力。

（4）灵活组网，简化运维。VLAN 能将不同地点、不同网络、不同用户组合在一起，形成一个虚拟的网络环境，就像使用本地 LAN 一样方便、灵活、有效，网络构建和维护更方便灵活，降低网络运行管理费用。

VLAN 的划分

（1）根据端口来划分 VLAN。按网络端口来划分 VLAN 配置过程简单，允许跨越多个交换机，缺点是灵活性差。

（2）根据 MAC 地址划分 VLAN。根据每个主机的 MAC 地址来划分，最大优点就是当用户物理位置移动时，VLAN 不用重新配置；但难应用在大规模 VLAN 中。

（3）根据网络层划分 VLAN。根据每个主机的网络层地址或协议类型划分，比如 IP 地址，但它不是路由，与网络层的路由无关。优点是用户的物理位置改变了，不需要重新配置所属的 VLAN，缺点是效率低。类似还有根据 IP 组播划分 VLAN 的方法。

（4）基于规则的 VLAN。基于策略组成的 VLAN 能实现多种分配方法，包括 VLAN 交换机端

口、MAC 地址、IP 地址、网络层协议等。当一个站点加入网络中时,被自动地包含进相应的 VLAN 中;同时对站点的移动和改变也可自动识别和跟踪,是最灵活的 VLAN 划分方法。其他还有按用户定义、非用户授权划分 VLAN 的方法。

参考文献

1. Tanenbaum A S. 计算机网络. 4 版. 潘爱民,译. 北京:清华大学出版社, 2004

2. IEEE Std. 802.1Q—2005. Virtual bridged local area networks. ISBN 0-7381-3662-X

(金耀辉)

xuni shengyin shengcheng

虚拟声音生成(virtual sound synthesis, VSS) **虚拟现实**的听觉表现方式。它为虚拟现实提供合成的声音信号,从而提高虚拟现实的真实感、**沉浸感**和交互性。虚拟声音的生成分为 3 个阶段,即定义—建模—重建。定义阶段给出虚拟听觉生成系统的先验信息,如声学环境尺寸、声源、听者参数等。建模分为 3 个方面:声源建模、传播介质(房间声学)建模和接收者(听者)建模。重建阶段利用建模的参数对声源进行加工,采用特定的输出设备,如耳机或扬声器,播放合成的虚拟声。

虚拟声音生成的关键是建模阶段。其中声源建模根据定义进一步确定声音所表达的信息,如声音内容、声音所处的空间、声源的个数、声源的运动状况、声源的方位以及接收者(听者)的运动状况等。在此基础上选择或产生声源信号,声源信号必须是理想的单声道信号,以避免在可听化过程中产生重叠的空间效应。

传播介质建模用于重建虚拟听觉环境,包括声信号在介质(空气)中的传播过程、空间中物体对声信号的反射、散射等作用。建模方法通常分为物理法和感知法。两种方法都是通过声学环境的脉冲响应函数(room impulse response, RIR)来描述听觉空间、声源与听者的方位特征。

接收者(听者)建模需要对接收者的方位信息进行建模,即模拟听者相对于声源信号的三维空间,包括方位角、仰角、距离。人耳听觉系统对声源方位的感知线索包含耳间时间差(interaural time difference, ITD)、耳间强度差(interaural intensity difference, IID)和包含频谱特征的耳廓效应,其中耳廓效应表明听觉系统对不同空间方向的声音频谱进行了增强或衰减,这一理论用于解释人的听觉系统对声源仰角方位的判断。

近年来,虚拟声音生成向多模态联合、声音质量可用性、联合现实等方向发展,其中包括声音与视觉显示的融合和相互作用,以及对虚拟现实感知的影响,声音与力触觉的集成;虚拟现实中虚拟声音的评价,包括声音质量、充分性和足够性;真实环境中虚拟声音生成的混合现实建模。以上虚拟声音生成的发展将有助于实现虚拟现实的多模态和联合体系。

参考文献

1. Begault D R. 3-D sound for virtual reality and multimedia. NASA/TM-2000-0000. 2000

2. Blauert J. Spatial hearing: the psychophysics of human sound localization. Revised ed. Cambridge, MA: The MIT Press. 1996

(吴镇扬 周琳)

xuni shijing shengcheng

虚拟视景生成(virtual scene synthesis) 利用计算机技术构建虚拟的三维场景,并通过显示设备进行具有真实视觉呈现的方法和技术。虚拟视景生成侧重于计算机生成结果的视觉真实性与实时性,可广泛应用于如虚拟训练、虚拟制造、虚拟军事演习、虚拟外科手术或虚拟城市规划等应用领域。实现虚拟视景生成,主要包含两类技术,一是虚拟世界数字模型的构建技术,二是将数字模型进行真实感视觉呈现的**计算机图形学的实时绘制**技术。

根据建模对象的不同,虚拟世界数字模型的构建技术主要包括几何建模、外观建模、运动建模、行为建模、物理建模等。几何建模是定量描述虚拟世界中事物的形状,可使用工具包、包含几何建模功能的动画/CAD 软件、三维数字化仪、三维扫描仪进行建模;外观建模是定量描述虚拟场景中的光源以及对象表面的材质和表面纹理;运动建模主要用于确定三维对象在世界坐标系中的位置变化和形状变化;行为建模是描述虚拟对象(角色)根据虚拟环境与自身状态进行主动行动的数学模型;物理建模定量描述了虚拟世界中对象的物理特征,如重量、惯量和硬度等。

将数字模型进行真实感视觉呈现,需要采用计算机图形学实时绘制技术(参见**实时绘制**)。实时绘制的要求是要在保证时间约束的前提下,尽可能地提高图形绘制的真实感。常见的实时绘制技术包括可见性剔除、层次细节、**基于图像的绘制**等。

现今,虚拟场景绘制主要有两个发展方向:一是面对诸如虚拟样机、仿真模拟器、军事仿真等需要大规模虚拟场景的应用需求,通过适当限制交互自

由度,并引入数据布局技术、可见性预计算、外存技术、并行分布计算技术、GPU 计算技术、实时光线跟踪技术等,寻找低复杂度、高真实性的虚拟场景的表示和驱动方法来达到目的;二是利用视频信息来直接描述现实环境,并将实拍视频与虚拟环境进行叠加混合,提高场景感知的真实性和实时性。

由于虚拟视景生成技术主要的应用领域对视景的真实性有较高的要求,这使得虚拟视景生成技术的难点主要在保证生成数据的准确性。准确性主要包含空间位置准确性与材质颜色准确性。为达到准确性要求,一是需要从虚拟世界数字模型的构建上,提高数字模型的准确性;二是需要从实时绘制方法上,提高呈现技术的准确性。

参考文献

1. Burdea G C, Coiffet P. 虚拟现实技术. 2 版. 魏迎梅,等译. 北京:电子工业出版社,2005

2. 石教英. 虚拟现实基础及实用算法. 北京:科学出版社,2002

(华炜)

xuni xianshi

虚拟现实(virtual reality,VR)　以计算机技术为核心,结合相关科学技术,生成与一定范围真实世界在视、听、触感等方面高度近似的数字化环境的有关技术、装置和理论。

对现实世界的景物进行模拟仿真,一直是人类追求的一个目标。现代科学技术的发展把人类的这一追求不断推向新的阶段和高度。虚拟现实所生成的数字化环境在模拟现实世界方向上达到了新境界。用户可以借助必要的装备与数字化环境中的对象进行交互作用,相互影响,产生亲临相应真实环境的感受和体验。例如,人们可以在虚拟战场环境中进行军事训练和演练,在虚拟人体上进行手术训练和手术规划,也可以乘虚拟光速航天器在虚拟太空漫游,体验相对论世界。

1965 年,**计算机图形学**的重要奠基人 Ivan Sutherland 发表了一篇短文"The Ultimate Display",以其丰富的想象力描绘了一种新的显示技术。他设想在这种显示技术支持下,观察者可以直接沉浸在计算机控制的虚拟环境之中,就如同在真实世界一样。观察者还能以自然的方式与虚拟环境中的对象进行交互,如触摸感知和控制虚拟对象等。Ivan E,Sutherland 的文章从计算机显示和人机交互的角度提出了模拟现实的思想,推动了计算机图形技术的发展,并启发了头盔显示器、**传感手套**等新型人机交互设备的研究。

1966 年,美国麻省理工学院(MIT)的林肯实验室研制出了第一个头盔式显示器,随后又将模拟力/触觉的反馈装置加入到系统中。总体上说 20 世纪六七十年代是 VR 思想、概念和技术的酝酿形成阶段。

进入 80 年代,随着计算机技术,特别是**个人计算机**和**计算机网络**的发展,VR 技术发展加快。这一时期出现了几个典型的虚拟现实系统,推动了虚拟现实技术发展。1989 年,Jaron Lanier 提出了 Virtual Reality 一词,很快这一术语被研究人员普遍接受,成为这一科学技术领域的专用名称。

90 年代以后,由于军事演练、航空航天、复杂设备研制等重要应用领域的巨大需求,VR 技术进入了快速发展时期。这一时期用于虚拟现实系统开发的软件平台和建模语言开始在市场出现。1994 年在日内瓦召开的第一届 WWW 大会上,提出了虚拟现实建模语言(virtual reality modeling language,VRML),开始了相关国际标准的制定。同年,Burdea G 和 Coiffet 出版了"Virtual Reality Technology"一书,用 3I(Immersion、Interaction、Imagination),即沉浸、交互和想象概括了虚拟现实的基本特征。

20 世纪 90 年代初我国一些高校和科研院所的研究人员从不同角度开始对虚拟现实进行研究,在虚拟现实理论研究、技术创新、系统开发和应用推广方面都取得明显成绩。由于虚拟现实的学科综合性和不可替代性,以及经济、社会、军事领域越来越多的应用需求,2006 年国务院颁布的《国家中长期科学和技术发展规划纲要》将虚拟现实技术列为信息领域优先发展的前沿技术之一。2008 年美国工程院公布了人类 21 世纪面临的 14 个重大工程挑战问题,虚拟现实是其中之一。

虚拟现实系统的分类

虚拟现实的研究对象和研究目标决定了它的学科交叉性和技术通用性,这使得其应用系统遍及社会各行业。从系统的功能和作用,也就是从应用的角度,虚拟现实系统可分为操作训练、规划设计和展示娱乐等几类。操作训练类系统广泛应用于各种危险环境(如核设施、水下设施)、作业对象难以获得(如医疗手术、航天器维修),以及耗费巨大(如军事演练)的行业领域的技术业务操作训练和效果评价;规划设计类系统用于新建设施、装备、作业方案、流程的演示验证与优化决策,可以大幅度降低设计成本,提高设计和作业方案的科学性、合理性。如城

市、社区、楼宇的规划设计、装备产品的虚拟设计与虚拟样机，以及军事行动和医疗手术的规划决策等；展示娱乐类系统将现实世界或假想世界场景数字化，供用户逼真地观赏体验，如虚拟景观、数字博物馆，以及各种游戏、影视制作等。

从沉浸式（参见**沉浸感**）体验角度，虚拟现实系统有非交互式体验、人-虚拟环境交互式体验和群体-虚拟环境交互式体验等几类。这种角度强调用户与虚拟环境的交互。对于非交互式体验来说，用户对虚拟环境的体验是被动的，体验的内容和流程是规划好的，不发生实质性交互行为，如场景漫游、三维影视等；在人-虚拟环境交互式体验系统中，用户可以通过交互设备（如传感手套、数字兵器、数字手术器械等）与虚拟环境进行交互，虚拟环境中的景物对交互行为做出实时响应，使用户能感受到虚拟环境的变化，从而产生对相应现实世界的体验，如飞行模拟器、单兵训练环境等；群体-虚拟环境交互式体验系统是人-虚拟环境交互式体验系统的多机化、网络化。多个用户共享一个虚拟环境，同时与包括用户化身在内的虚拟环境进行交互，感受到和虚拟环境及其他用户相互作用的体验，如军事仿真演练、网上交互游戏等。

虚拟现实存在的主要问题

虚拟现实是新兴的信息科学技术领域，研究内容包括虚拟环境构造和应用的各个方面，存在的科学技术问题主要集中在建模、表现和人机交互三个方面。

1. 虚拟现实中的建模　建模是虚拟现实的核心问题。现实世界对象在计算机数字空间中的表达，需要通过数学、物理，以及各种数字化方法将其自身状态、与其他对象之间的关系与相互作用及动态变化所遵循的规律映射为数字空间中的各种动静态数据表示，这一过程称为建模。建模存在一系列基本理论问题有待解决，如事物的可建模性、模型的复杂性、相似性，特别是模型的可信性度量等。

虚拟现实建模方法与所要模拟的对象类有关，也与应用领域密切相关。针对所要模拟对象的不同方面，可将建模方法分为景物外观建模、基于物理的建模、行为建模和虚实融合建模等。景物外观建模主要包括基于点云数据、网格、体素和材质光照信息的建模方法。景物外观建模起步时间较早，研究较多。物理建模在于体现对象的物理特性，使得虚拟环境中的动态景物更加逼真。目前体现较多的是虚拟实体的运动学和动力学特性，为进一步提高真实感，关于虚拟实体的材料、弹塑性和黏稠性等物理特性的研究越来越受到重视。物理特性的计算模型、高效算法以及有效的表现方法是需要研究解决的三大问题。近年来，虚拟现实应用对自治对象行为的智能水平提出了越来越高的需求，行为建模成为虚拟现实研究中一个重要方面。这一领域总体上属于**人工智能**的研究范畴。随着虚拟现实应用领域的不断扩展，虚实融合的环境受到人们的重视。通过虚拟现实建模构造相应的虚拟景物，将其融入真实场景，可以有效扩展人机交互的空间和虚拟现实的应用领域。其主要问题有虚实对象三维注册、遮挡处理、虚实光照融合处理等。由于现实世界形态的复杂性、发展的无限性，以及人类认识世界的局限性，使得建模成为虚拟现实领域永恒的研究内容。

2. 虚拟环境对象的逼真表现　表现技术是虚拟现实的重要研究内容。它将数字空间内的各种对象模型通过一定的表现方法和算法渲染在表现设备上呈现给用户，使用户产生相应的视、听、触（力）和嗅（味）觉等感受体验。表现方法与对象模型和表现设备直接相关，而且各种表现设备，如透视式头盔、全息真三维显示、力反馈操纵和触觉数据手套等都在不断发展，因此，这也是一个永恒的问题领域。嗅觉是人体的重要感知，如何让人感受到各种气味是虚拟现实领域研究的一个内容。嗅觉是由化学刺激产生的，这和视、听、触觉有很大不同，是一个难度很大的领域。

3. 人机交互　主要是人对虚拟环境中对象的操作控制。这类问题涉及人或外部世界与虚拟环境对象之间互相作用的信息交换方式和人机交互设备。虚拟现实人机交互与计算机系统的人机交互具有十分密切的关系，既有一些共同的方式，也有明显的独特之处。计算机系统的人机交互主要是人与计算机之间的信息传递，而虚拟现实人机交互更强调人与虚拟环境之间的感知传递，这要求更高的自然性和真实感。真三维显示、6 自由度鼠标、三维空间方位跟踪，以及新概念虚拟现实人机交互机制和大量的面向领域应用的专用人机交互机制和人机交互设备是虚拟现实人机交互的重要研究内容。

虚拟现实是一个充满活力和挑战的科学技术领域，不仅存在有待解决的问题和难题，而且还在不断产生新的理论和关键技术问题，这些问题的解决会导致虚拟现实技术与应用的巨大发展。

参考文献

1. Sutherland I E. The ultimate display. In：Pro-

ceedings of the International Federation of Information Processing (IFIP) Congress, 1965: 506-508

2. Grigore B, Phillippe C. Virtual reality technology. 2nd ed, John Wiley and Sons, 2003

3. Zhao Q P. A survey on virtual reality. Science in China Series F (Information Sciences), 2009, 52(3): 348-400

(赵沁平)

xuni xianshi jiaohu shebei

虚拟现实交互设备(interaction devices of virtual reality) 指在**虚拟现实**系统中实现操作者与计算机系统相互作用的各种硬件装置(包含输入设备与输出设备),其服务对象是操作虚拟现实系统的用户。虚拟现实交互设备的主要功能一方面是获取物理世界的各种动态信息,另一方面则是通过特殊装置实现对某一物理过程的模拟,并使得操作者感受到这种物理过程。虚拟现实交互设备主要有:

1. **位置跟踪器**
2. **传感手套**
3. **力触觉反馈装置**
4. **运动捕获系统**
5. **立体显示装置**

参考文献

1. 石教英. 虚拟现实基础及实用算法. 北京:科学出版社,2002

2. Lepetit V, Moreno-Noguer F, Fua P. EPnP: an accurate $O(n)$ solution to the PnP problem, International Journal of Computer Vision, 2009, 81(2)

3. Reichlin F, Leibe B, Koller-Meier E, et al. Online multiperson tracking-by-detection from a single, uncalibrated camera. IEEE Trans on Pattern Analysis and Machine Intelligence, 2011, 33(9): 1820-1833

(王涌天 刘越)

xuni zhongduan

虚拟终端(virtual terminal, VT) 将计算机网络中各种类型的实终端的功能一般化、标准化后得到的一种终端模型。网络中的应用程序和实终端经过映像后,变换成标准的虚拟终端进行通信。

虚拟终端模型由3个部分组成:

(1) 控制部件 实现虚拟终端协议(VTP),并实现与传送级服务接口。

(2) 数据结构 包括正文数据及其属性,并记录用户与应用程序之间通信的当前状态。不同类型的虚拟终端所需的数据结构不同。

(3) 终端处理程序 提供虚拟终端与本地设备(如输入、输出等设备)之间的接口。

由于计算机终端种类繁多,各种实际应用中对终端的要求也不同,要想将所有终端的众多功能归结在一起,抽象化、一般化为一个虚拟终端模型是很困难的。即便如此,也不利于终端功能的扩充。一个有效的方法是按类建立虚拟终端模型及其协议。国际标准化组织将虚拟终端分成5类:

(1) 基本类 处理由字符元素或镶嵌元素组成的一维、二维或三维数组,对应的实终端有电传打字机、页式显示终端等;

(2) 表格类 处理对象与基本类相同,另外还具有对输入、输出进行表格式控制处理的能力,对应的实终端有处理各类业务(如航运、银行等)的窗口服务终端;

(3) 图像类 处理由像元元素组成的二维或三维数组,对应的实终端是图像处理终端;

(4) 图形类 处理由几何图形元素(如点、线、二次曲线段等)组成的多层结构,对应的典型实终端是图形终端;

(5) 混合类 处理由字符元素、镶嵌元素、像元元素和几何图形元素组成的多层结构,它是以上4类的混合,对应的典型实终端是高级工作站。

为了提供网络环境下使用分时系统的功能,一般要在提供服务的远地主机处引入虚拟终端,并在其分时系统中配置相应的虚拟终端控制程序。这种虚拟终端实际上是由虚拟终端协议服务进程实现的,它与实际物理终端无关,完全是按虚拟终端协议的规定,进行公共处理而产生所需的控制信息的数据代码。另一方面,为了便于用户通过自己的物理终端或用户程序去使用远地分时系统,则启动虚拟终端协议使用进程,按照虚拟终端协议的规定,把本地系统所支持的终端映象成标准虚拟终端,其功能包括与远地主机之间建立或拆除一组连接、传输控制信息和数据信息、产生中断信号等。这样,用户就能通过自己的终端访问远地分时系统,传送数据和文件内容,控制作业运行,调用各种服务子系统。ARPA网的虚拟终端协议称为Telnet,是针对基本类终端设计的。

参考文献

1. 曹东启,等. 计算机网络软件基础. 北京:人民邮电出版社, 1982

2. 胡道元. 计算机局域网. 北京:清华大学出

版社, 1990 (李学农)

xuni zhuanyongwang

虚拟专用网(virtual private network, VPN)

通过公用的网络架构(如Internet),将物理分散的网络连接成的逻辑专用网,常用于连接大中型企业、团体私有网络,主要采用隧道、加密和认证等技术保证信息的私密性和完整性,防止信息被窃取和篡改。

VPN有多种划分方法:

(1)按VPN的应用方式分类 VPN的应用方式包括内联网(Intranet)VPN、外联网(Extranet)VPN和远程接入(Access)VPN。内联网VPN通过用户自己的网络架构实现用户内部网络各局域网的安全互联;外联VPN将若干个企业的Intranet VPN结合起来构成一个大的虚拟企业内部网络;远程接入VPN使用公共网络作为骨干网在设备之间传输VPN的数据流量,利用公共网络的拨号及接入网(比如PSDN和ISDB)实现虚拟专用网。

(2)按所用的设备类型分类 根据使用的设备,可以将VPN分为路由器式、交换机式和防火墙式三类。路由器式VPN适用于广域网,实现不同地域之间VPN网段的互联,目前主流的实现技术是基于多协议标记交换协议(MPLS),视互联的VPN网段类型而分为链路层VPN和网络层VPN,即L2VPN和L3VPN。交换机式VPN基于802.1q的VLAN技术,主要应用于在一个以太交换网中实现使用户群分隔的VPN网络。防火墙式VPN是最常见的一种VPN的实现方式,提供两点之间的安全通道。

(3)按VPN的隧道协议分类 传统VPN的隧道协议主要有PPTP、L2TP和IPSec三种。PPTP、L2TP是第2层隧道协议,都是将数据封装在点对点协议(PPP)帧中通过互联网发送。IPSec是第3层隧道协议,通过对IP报文的加密保护来实现隧道传输。

随着技术的发展,出现了一些新的VPN解决方案,如MPLS VPN、SSL VPN和Socks5 VPN。

MPLS VPN在网络路由和交换设备上应用MPLS技术,简化核心路由器的路由选择方式,利用传统路由技术的标记交换实现的IP虚拟专用网络。MPLS VPN能够利用公用骨干网广泛而强大的传输能力,降低企业内部网络的建设成本,极大地提高用户网络运营和管理的灵活性,同时能够满足用户对信息传输安全性、实时性、高带宽、方便性的需要。

SSL VPN是采用SSL(Security Socket Layer)协议来实现远程接入的一种新型VPN技术,运行在传输层,包括服务器认证、客户认证(可选)、SSL链路上的数据完整性和保密性。SSL VPN在访问控制方面具有更细的粒度,与IPSec VPN只搭建虚拟传输网络不同的是,SSL VPN重点在于保护具体的敏感数据,可以根据用户的不同身份,给予不同的访问权限。相对于传统的IPSec VPN而言,SSL VPN具有部署简单、无客户端、维护成本低、网络适应性强等特点。

Socks 5 VPN是在总结IPSec VPN和SSL VPN优缺点的基础上提出的解决方案。Socks 5 VPN运行在会话层,对应用数据和应用协议具有可见性,网络管理员能够对用户远程访问实施细粒度的安全策略检查。Socks 5 VPN采用瘦客户端和服务器架构,在用户访问远程资源之前执行相应的身份认证和访问控制,只有通过检查的合法数据才允许流进应用服务器,从而有力保护了组织的内部专用网络。

基于Internet的VPN为企业和组织提供了安全、可靠的Internet访问通道。VPN具有高安全性、可扩展性、简单灵活、成本低廉、易于管理等特点,已被大量应用于企业网。

参考文献

1. Tanenbaum A S, Wetherall D J. Computer networks. 5th ed. London: Prentice Hall, 2010

2. 高海英,等. VPN技术. 国家信息化安全教育认证(ISEC)系列教材. 北京:机械工业出版社, 2004

(吴纯青)

xuqiu dingyi yuyan

需求定义语言(requirements definition language) 用于书写**软件需求定义**的语言。软件需求包括**功能需求**和**非功能需求**两个方面。功能需求从用户角度明确了软件系统必须具有的功能行为,它是整个软件需求的核心所在。在功能需求的基础上,非功能需求对软件需求作进一步的刻画,它包括功能限制、设计限制、环境描述、数据与通信规程和项目管理等。软件需求定义主要面向用户,采用基于现实世界的描述模型,以便于用户理解。

需求定义语言的研究受计算机应用发展的推动,其发展大致可分为三个阶段。

从20世纪60年代末期到70年代初期为第一阶段。在计算机发展早期,由于问题规模较小,需求定义语言研究的重要性并未引起足够重视,人们常采用自然语言来书写一些简单的需求。自然语言对

于规模较小的应用还能够应付,但对于大型软件系统,其内在的非形式性导致需求定义中经常出现错误,并且由于由机器自动处理自然语言的非形式性异常困难而使得纠正需求定义中的错误不仅代价高而且费时。随着计算机应用规模的不断扩大,特别是自1968年在大西洋公约学术会议上提出软件工程以来,人们逐渐认识到需求定义语言的重要性,开始研究各种类型的需求定义语言。

从70年代初期到80年代中期为第二阶段。由于认识到非形式自然语言给需求定义带来的种种不足,人们以**软件方法学**为基础,开始研究需求定义语言的形式化问题。提出了诸如基于自顶向下途径的SA,基于自底向上途径的问题陈述语言PSL,以及基于面向对象思想的需求定义语言RML等,这类语言的重要特征是较自然语言有比较精确的语法和语义定义,从而便于分析其各种性质,利于用计算机提供自动化的支持。

80年代中期迄今为第三阶段。随着软件系统复杂性的提高和规模的增大,需求定义愈加困难和耗时;传统的需求定义语言由于采用的模型和问题领域差距较大,从而使得需求定义的易理解性和易维护性较差。而面向对象模型由于和问题结构之间有良好的对应关系而较好地满足了需求定义的需要。因此,面向对象需求模型的研究成为热门课题并展示出良好的前景,代表性的工作有J. Rumbaugh等提出的模型以及D. W. Embley等提出的模型。这些模型以对象及其相互间的关系为核心,以图形化表示机制为手段来刻画系统,不仅能反映系统的静态结构关系,而且能反映系统的动态变化行为,为解决传统功能分解模型中存在的功能本身的易变性、分解结构的随意性以及功能结构与问题结构常难对应等问题提供了有效途径。

按照形式化的程度,需求定义语言可分为非形式需求定义语言,半形式需求定义语言以及形式需求定义语言三类。

非形式需求定义语言是指未作任何限制的自然语言。一般说来,非形式需求定义语言具有易理解、易使用的特点,易于为一般用户所接受。但是由于自然语言的非形式性而使得需求定义中常出现错误,并且难以用计算机系统提供自动化的支持。半形式需求定义语言是指在宏观上对语言的语法和语义有较精确的描述,而在某些局部方面则允许使用非形式的自然语言。这类语言既便于用户的表达和理解相应的需求定义,又可在某种程度上用机器对相应的需求定义进行管理和进行部分的正确性检查。形式需求定义语言是指其语法和语义均为精确定义的语言。一般说来,形式化需求定义语言具有良好的数学基础,易于分析需求定义的各种性质。然而,形式需求定义语言往往要求用户具有较高的数学修养,且相对说来,所书写的需求定义较难理解。目前,形式需求定义语言的研究正在进行之中,并且在非功能需求的形式化方面进展缓慢。

需求定义语言的研究主要涉及基本模型、**语言结构**和**语用分析**等。

需求定义语言的基本模型是相应软件开发风范在语言中的具体体现。它是表达软件需求的基础,决定了参加需求分析的各类人员的思维方式。两类重要的模型分别是功能分解模型和面向对象模型。在功能分解风范下,语言的用户通常将待解的问题抽象成需要满足的功能,通过功能分解的方法来表述系统各个部分之间的关系,以此为架构来刻画用户对系统的需求。而在面向对象风范下,语言的使用者可以比较直接地刻画现实世界模型,并在此基础上描述对所需软件的需求。从而使得相应需求定义易于理解、易于修改和易于复用。不同的基本模型往往适合于不同的应用领域,并对语言结构的选取以及相应的支撑方法产生重要影响。

语言结构提供了用于刻画系统需求的具体手段,主要包括基本模型的描述、数据描述、控制描述、抽象机制以及项目相关信息描述等,当然,不同语言的侧重点有所不同。在功能分解模型的描述方面,SA采用自顶向下、逐层分解的功能模型,为了支持这一模型的描述,SA提供了分层的数据流图。而PSL则采用了自底向上的途径,它提供了各种实体类型及其相互间的关系用以分别描述各个需求,然后形成所需的需求定义。当然,并不是所有语言均基于单一模型的单一方法,例如,RSL试图将自顶向下和自底向上的途径加以结合以适应实时系统描述的需要,它提供了R网等设施。在面向对象模型支持方面,通常有两种途径,其一是设计新的需求模型和语言来支持面向对象方法,例如,D. W. Embley等提出的面向对象需求模型,提供了各种图形化表示机制来刻画对象及其相互间的关系。其二是用已有语言来刻画面向对象模型,例如,ROOA方法采用形式规约语言LOTOS中的抽象数据类型和进程定义来描述面向对象需求定义,并对继承机制提供支持。在数据描述方面,PSL提供了ENTITY,CONSISTS OF,DERIVEDBY等实体或关系来描述数据对象的

名、数据结构和数据流程。在控制描述方面,RSL 提供了各种类似于程序设计语言中使用的控制结构。尽管此类结构对于通信系统和实时系统较为有用,但可能导致与实现有关的问题。在抽象机制方面,RML 语言在面向对象的架构下提供了聚合、分类和泛化三类抽象机制,它们可一致地用于语言中的三类规约单位:对象、活动、断言。在项目相关信息描述方面,各类半形式化的语言均提供了描述这些信息的手段。例如,PSL 可描述文档信息,这些信息通常是借助于自然语言来刻画的。

语用分析主要讨论需求定义语言的适用领域、可扩展性等性质,以及相应的方法与工具支持。例如,PSL 主要适合于商业应用,RSL 主要用于实时系统;而有些语言如 SA 则是通用语言;可扩展性指是否可在原有语言结构基础上从语法和语义角度定义新的语言结构;在此意义下,PSL 和 RSL 本质上是可扩展的。为了充分发挥需求定义语言的作用,必须研究与之相应的方法和工具。方法应给出获取需求的原理和步骤以及如何从需求开发相应的程序。而工具支撑则对需求的分析提供自动支持。在某种意义下,几乎所有语言均对如何形成需求提出建议和提示,但真正的方法却不多见。就工具支撑而言,SA 是为手工使用而设计的,而 PSL,RSL 则一开始就将工具支撑和语言设计集成在一起。

概括起来,需求定义语言的研究已取得较大进展,已有的各种类型的需求定义语言已逐步应用于软件工程实践并取得良好效果。近年来,国际上又兴起需求工程的研究热潮,1993 年召开了第一届需求工程国际研讨会,1994 年召开了第一届需求工程国际会议,国际信息处理协会 IFIP 成立了需求工程工作组,即 IFIP WG 2.9。需求定义语言是需求工程的核心内容之一,因此,可以预言,需求工程的研究必将促进需求定义语言的进一步发展。

参考文献

1. McDermid J A(ed). Software engineer's reference book. Butterworth-Heinemann Ltd. ,1991

2. Partsch H A. Specification and transformation of programs. Springer-Verlag, 1990

(吕建)

xuqiu gongcheng

需求工程(requirements engineering)　系统工程和软件工程中的一个子学科,研究用于发现、纪录、分析和维护对系统(特别是软件系统)的要求的方法和技术。

发展简史

"需求工程"这个术语最早出现在美国国防和空间系统研究组 1979 年的一篇技术报告中,但这篇报告并没有涉及需求工程的具体内容。1990 年,IEEE 计算机学会出版了第一部关于需求工程的教材,对需求工程的研究动机、研究问题和一些具体的技术进行了比较全面的总结。随之出现了需求工程的系列国际会议。

在软件工程领域,需求工程是开发软件需求的工程。在早期的软件开发模型中,需求工程处于软件开发的第一个阶段,这个阶段的制品是关于待开发软件产品的需求文档或软件需求规格书。在具有迭代特征的软件开发模型中,需求工程还作为软件系统的整个生命周期过程中的需求变化和更新的管理机制,负责记录软件系统在整个生命周期中的变更管理任务。需求工程的目标是,采用系统化的方法和工程化的管理手段,高效地开发出准确表达用户需求的软件规格说明。从这个意义上说,需求工程是**软件工程**前期的一个重要阶段,它的成功实施是软件的后续开发过程得以成功的先决条件。

近年来,需求工程得到长足的发展,从其发展进程和未来的发展趋势来看,需求工程已经不仅仅局限于实现软件工程为其设定的目标,它更强调要用一种自然的方式,去系统地寻找和发现正确的和潜在的用户需求,建立恰当的系统需求模型,并在对模型进行正确性分析和有效性验证的基础上,为软件开发的后续阶段提供既正确又合理的软件规格说明。

基本内容

需求工程的主要任务是,从客户和潜在用户对系统的需要和愿望中,抽取出可以实现的对软件的能力要求和约束,形成软件规格说明。其中,前者一般是不完整的,有时是含糊的,隐含了矛盾的,并且是用非形式的术语来表达的,而后者必须是完整的、精确的和一致的,最好可以用形式化的表示方法表示出来。这是软件开发过程中的关键活动,因为在这个活动中引入的错误是软件开发过程中维护代价最高的错误。它也是软件开发过程中最困难和需要审慎的社会活动,因为它需要需求相关者们的密切合作和沟通,而这些需求相关者可能具有完全不同的知识背景和不同的关注点。

为了完成上述任务,需求工程进行由如下三个活动组成的系统化过程,即通过迭代合作的问题分析过程去开发需求,获得必要的软件需求信息;采用

不同的建模方法和表示技术将所得到的软件需求信息文档化，形成软件规格说明；最后检查所获得的软件规格说明的准确性、正确性和完备性。

不同的问题分析视角产生了不同的需求工程方法，典型的方法包括：面向目标的方法，基于主体和意图的方法，问题框架方法，基于情景的方法，等等。贯穿这些方法和技术的核心是认识和观察现实世界的视角以及人类处理复杂问题的三个通用原则，即抽象、分解和投影。这些需求工程方法一般通过结构化的需求抽取、系统化的需求建模、形式化的需求验证、规范化的需求管理等机制，来降低需求工程过程的开销，并提高软件规格说明的质量。

软件工程研究如何用软件技术去解决现实世界的问题，而需求工程则是定义软件所需要解决的问题。现实的情况是，大部分软件开发所面对的问题几乎总是定义不足的，需求工程一般是要从定义不足的现实世界问题中，标识出需要解决的问题，并为问题建模，进一步地还需要从中找出软件能解决的问题部分，并确定要开发的软件的能力需求和约束。需求工程要使得构造出来的软件能力需求和约束满足两方面的评判标准，一是要保证构造出来的软件需求是可实现的，即需求的正确性；二是要保证按照这个需求模型实现出来的软件确实是用户想要的，即需求的有效性。

需求工程的主要难点包括：

（1）克服知识的鸿沟　需求工程的出发点一般来自其他行业的领域问题，由于知识鸿沟存在，需求工程的首要任务之一，就是与各类需求相关者（包括其他行业的专家和最终用户等）进行信息的沟通与知识的传递。由于各类需求相关者具有不同的知识背景，对问题本身有不同的着眼点，沟通上的困难使软件需求工程的实施增加了人为因素的难度。

需求相关者按其在需求工程活动中的主要任务可以分为两类，一类是信息提供者，包括领域专家、领域用户、系统投资方等。另一类是信息接收和建模分析者，包括系统分析员和软件需求分析员。需求分析员的任务是理解领域问题，并从领域问题中识别出软件技术可以实现的部分，构造软件问题模型并进行分析，保证软件问题描述的正确性。除了知识背景不同带来沟通上的困难外，知识的鸿沟更主要地体现在，需求分析员一般而言只具备软件领域知识，但为了要完成理解领域问题的任务，他们常常需要学习大量的领域知识，成为问题领域的专家。

（2）涉及多学科的交叉　需求工程具备固有的学科交叉性。首先，它研究的对象是系统，系统论以及系统科学和工程方法是问题建模的认识论基础。其次，需求工程是关于建模的，它从现实问题出发，构建良定的软件模型，基于数学和逻辑的形式化建模技术以及基于软件技术的结构化建模方法都是必要的研究手段。更广泛意义上的学科交叉性体现在对需求工程师的个人能力要求和素养上，涉及掌握研究如何与人打交道的社会科学的知识，比如，人类学和民族方法学、组织行为学、社会心理学、政治学等，掌握研究如何认识世界的认知科学的知识，比如，认知心理学、语言学、知识表示等，以及掌握研究如何观察世界的哲学学科的知识，比如，观察实验法和科学的哲学、现象学、认识论和本体等。

（3）实现从非良定问题到良定问题的转换　人对现实世界观察常常是不完整的，观察的结果还可能存在不确定性，或者本质上就是不确定的，同时，矛盾的知识会以不同的形式出现在不同的上下文中，这些在现实世界中都不需要事先进行特别的处理，一般等到事情发生再有针对性地解决。但软件系统是一个形式系统，一旦确定后，机器只会照章办事，因此必须确保软件系统的需求描述是完备、确定和无矛盾的，否则将会导致软件系统运行时刻错误，严重影响系统的可靠性。

（4）同时保证正确性和有效性　需求工程的另一个困难在于判断得出的需求规格说明是否正确和是否有效。需求模型的正确性问题是，所提出的对软件系统的要求和约束是否当前可实现的。当软件需求足够精化到可以用已有的形式化方法建模的时候，可以用已有的形式化验证工具判断需求模型的正确性。需求模型的有效性问题是，系统需求是否真正表达了现实世界的问题以及由此产生的解决方案是否客户真正想要的。针对需求的有效性问题，除了让用户待软件系统开发出来进行确认外，目前还没有更好的办法解决。

展　望

目前，需求工程的研究和实践还处于百家争鸣、百花齐放的时代，各种不同的方法以其自己独有的视角解决需求工程中的问题。随着软件问题规模的不断扩大，软件和现实世界不断地融合，需求工程的研究开始从只关注系统功能的建模和分析，过渡到重点关注系统质量需求的建模和分析以及其他为应对开放环境带来的一系列软件问题而引出的其他非功能性约束的建模和分析。可以预计，需求工程的研究和实践还将进一步繁荣。

参考文献

1. 金芝,刘琳,金英. 软件需求工程:原理和方法. 北京:科学出版社,2008

2. van Lamsweerde A. Requirements engineering: from system goals to UML models to software specification, Wiley, 2009

3. Alfor M W, Lawson J T. Software requirements engineering methodology (Development). TRW Defense and Space Systems Group, 1979

4. Thayer R H, Dorfman M. (eds.) System and software requirements engineering. Los Alamitos, CA: IEEE Computer Society Press, 1990 (刘超)

xulie wajue

序列挖掘(sequence mining) 对序列数据的挖掘(参见**数据挖掘**)。序列是按一定次序排列的对象(或事件)。分为时序数据和序贯数据。时序数据是由随时间变化的值或时间组成的数据,而序贯数据是由有序事件组成的数据。按处理的序列不同和所挖掘的模式不同,可将序列挖掘粗略地分为:序列建模、序列分类和聚类、序列趋势分析、序列模式挖掘等。

由于序列数据具有高维性、动态性、大规模、噪声高等特性,通常需要对序列数据进行预处理,即序列建模,才能得到有效的挖掘结果。因此,序列建模是序列挖掘的基础,它的任务是对原始的序列数据进行一定的抽象和概括,以较好地提取出序列数据的代表特征。目前,序列建模方法主要包括:频域表示法(将时序数据从时域通过傅里叶变换或小波变换映射到频域,用低频系数来代表原来的数据);分段线性表示法(用若干个直线段来近似代替原来的序列数据);符号表示法(用离散化方法将序列数据的连续实数值或者一段时间内的时序波形映射到有限的符号表上);主成分分析表示法(对整个序列数据集进行分析,抽取数据集上的主要特征来表示原来的序列数据)等。

序列分类是在有类标的序列数据集上训练一组分类器,使得它能对未来的数据进行分类。序列聚类是将序列数据集自动划分为若干类,使得类内的序列间的相似性尽可能地大、类间的序列间的相似性尽可能地小。因此,序列间的相似性计算常常是序列分类和聚类算法的基础。目前常用的序列相似性计算方法有:Lp 距离,动态时间弯曲距离,最长公共子序列,编辑距离,串匹配等。较之一般数据集上的分类和聚类问题,序列分类和聚类相对更困难,其主要原因在于要处理的序列数据不等长。即使是等长的序列,由于不同序列在相同位置的数值一般不可直接比较,使得一般的分类和聚类算法不能直接应用于所要处理的数据。因此,需要针对具体的应用定义合适的相似性度量,并对一般数据集上的分类和聚类方法进行改进,才能得到较好的分类和聚类性能。

简单地说,序列趋势分析是预测序列数据的未来值,包括以下四种类型:①长期或趋势变化(Trend Movement);②循环运动或循环变化(Cyclic Movement or Cyclic Variations);③季节性运动或季节性变化(Seasonal Movements or Seasonal Variations);④非规则或随机变化(Irregular or Random Movements)。常见的趋势分析方法主要是回归分析法和机器学习方法。

序列模式挖掘包括序列频繁模式挖掘和序列异常模式挖掘两种。序列频繁模式挖掘是挖掘出序列集中频繁出现的有序事件或子序列。通常是将传统的基于事务数据库的**频繁模式挖掘**方法(如 Apriori 方法、FP growth 方法等)扩展到序列数据集上,可用于识别生物序列中的功能元件(如 DNA 序列中的顺试调控元件、RAN 序列中的选择性剪切位点等)。序列异常模式是在一个序列中与其他模式比较具有显著差异的模式。目前已经提出多种序列异常模式检测方法,如基于人工免疫系统的检测法,基于支持向量的检测法,以及基于马尔可夫模型的检测法等。

由于序列数据大量存在于生产和生活实践之中,序列挖掘在经济、金融、工程、生物、医学等众多领域有着广泛的应用。

参考文献

1. Han J, Kamber M, Pei J. Data mining: Concepts and techniques. 3rd ed. Morgan Kaufmann Publishers, 2011

2. Dong G, Pei J, Sequence data mining, Springer, 2007 (贾彩燕 黄厚宽)

xushu

序数(ordinal number) 良序集(参见**二元关系**)的序型。若令每个线性序集 M 都对应于唯一的一个符号 σ_M,使得对任意两个线性序集 M_1 和 M_2 皆有 $\sigma_{M_1}=\sigma_{M_2}$ 当且仅当有从 M_1 到 M_2 的序同构,则称 σ_M 为 M 的序型,并常用 $\overline{M}$ 表示之。

$\langle \mathbf{N},\leqslant\rangle$ 的序型用 ω 表示。

有穷良序集的序型称为有穷序数，全部有穷序数显然为 0，1，2，…。无穷良序集的序型称为超穷序数。ω 就是一个超穷序数。

序型加法　设 σ 和 τ 为两个序型，所对应的线性序集分别为 $\langle A_1, R_1\rangle$ 和 $\langle A_2, R_2\rangle$，不妨假定 $A_1 \cap A_2 = \varnothing$。若令 $R = R_1 \cup R_2 \cup (A_1 \times A_2)$，则 R 为 $A_1 \cup A_2$ 上的一个线性序。通常称线性序集 $\langle A_1 \cup A_2, R\rangle$ 的序型为 σ 与 τ 的和，记为 $\sigma + \tau$。显然此定义与 $\langle A_1, R_1\rangle$ 和 $\langle A_2, R_2\rangle$ 的具体取法无关。

因为 $1 + \omega = \omega$ 且 $\omega + 1 \neq \omega$，所以序型加法不可交换，但可结合，即对任意序型 σ，τ 和 λ 恒有

$$(\sigma + \tau) + \lambda = \sigma + (\tau + \lambda)$$

故而此和可记为 $\sigma + \tau + \lambda$。

序型乘法　设 σ 和 τ 为两个序型，所对应的线性序集分别为 $\langle A_1, R_1\rangle$ 和 $\langle A_2, R_2\rangle$，不妨假定 $A_1 \cap A_2 = \varnothing$。若在 $A_1 \times A_2$ 上定义二元关系 R 如下：

$\langle\langle x_1, y_1\rangle, \langle x_2, y_2\rangle\rangle \in R$　　当且仅当

$\langle y_1, y_2\rangle \in R_2$ 或 $y_1 = y_2$ 且 $\langle x_1, x_2\rangle \in R_1$

则 R 显然为 $A_1 \times A_2$ 上的一个线性序。称线性序集 $\langle A_1 \times A_2, R\rangle$ 的序型为 σ 与 τ 的积，记为 $\sigma \cdot \tau$。显然此定义与 $\langle A_1, R_1\rangle$ 和 $\langle A_2, R_2\rangle$ 的具体取法无关。

因为 $2\omega = \omega$ 且 $\omega 2 = \omega + \omega \neq \omega$，所以序型乘法不可交换。但序型乘法是可结合的，即对任意三个序型 σ，τ 和 λ 恒有

$$(\sigma\tau)\lambda = \sigma(\tau\lambda)$$

故而此乘积可记为 $\sigma\tau\lambda$。

序型乘法满足第一分配律，即对任意三个序型 σ，τ 和 λ 恒有

$$\sigma(\tau + \lambda) = \sigma\tau + \sigma\lambda$$

因为 $2(\omega + 1) \neq (\omega + 1)2$，所以序型乘法不满足第二分配律。

设 σ 为任意序数。若有序数 τ 使 $\sigma = \tau + 1$，则称 σ 为***后继序数***或非极限序数，否则称 σ 为***极限序数***。有穷序数和 $\omega + 1$ 都是后继序数，ω 与 $\omega + \omega$ 都是极限序数。

序数的幂　设序数 $\sigma \neq 0$，则 σ 的幂定义如下：

(1) $\sigma^0 = 1$；

(2) 若 τ 为后继序数，即有序数 λ 使 $\tau = \lambda + 1$，则 $\sigma^\tau = \sigma^\lambda \cdot \sigma$；

(3) 若 τ 为极限序数，则 $\sigma^\tau = \lim\limits_{\xi < \tau} \sigma^\xi$。

（王兵山）

xunzhi fangshi

寻址方式（addressing mode）　生成计算机指令中实际使用的有效地址的方法。

不同的计算机有多种寻址方式，下面介绍广泛采用的几种，并以一地址指令为例加以说明。

(1) 直接寻址　指令的地址码部分给出操作数在存储器中的地址，如图 1 所示，图中的寻址方式由操作码 OP 约定，A 为地址。

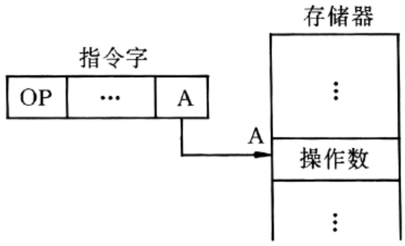
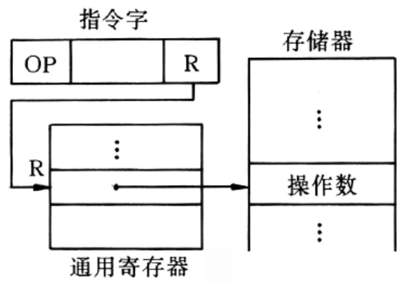


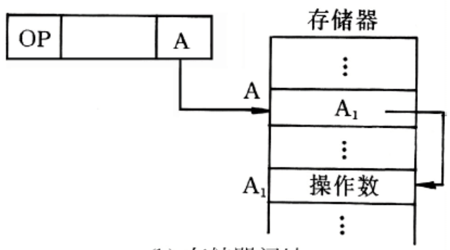
图 1　直接寻址

(2) 寄存器寻址　中央处理器中一般设置有一定数量的通用寄存器，用以存放操作数、操作数地址或运算的中间结果。指令的地址码 A 为通用寄存器地址，操作数在该寄存器中。这种寻址方式也可看作是一种直接寻址。

(3) 间接寻址　根据指令的地址码从存储器或寄存器中取出的内容不是操作数，也不是下一条要执行的指令，而是操作数的地址或指令的地址。这种寻址方式称为间接寻址，简称间址。间接寻址有一次间址和多次间址两种情况，大多数计算机只允许一次间址，图 2(a) 与图 2(b) 分别表示寄存器一次间址和存储器一次间址，图 2(b) 取操作数需访问存储器两次。



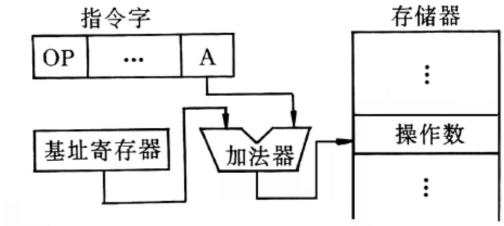
(a) 通用寄存器间址



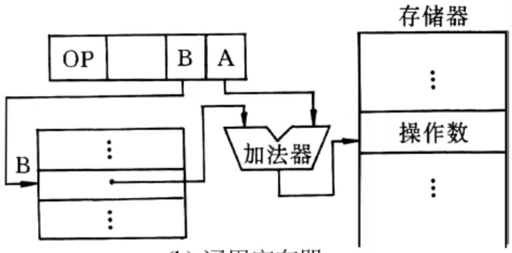
(b) 存储器间址

图 2　间接寻址

（4）基址寻址　在中央处理器中设置一个专用的基址寄存器，或由指令指定一个通用寄存器作为基址寄存器，操作数的地址由基址寄存器的内容和指令的地址码 A 相加得到，如图 3 所示。在这种情况下，地址码 A 实际上是一个位移量。



(a) 专用寄存器



(b) 通用寄存器

图 3　基址寻址

基址寄存器主要用于为程序或数据分配存储区，或扩大地址码长度。通常基址寄存器中的内容只能由系统程序设定（通过特权指令实现），不能被用户程序所修改，因此确保了系统的安全。

图 3（b）指令字的地址码固定通用寄存器的地址。

（5）变址寻址　操作数地址由变址寄存器内容和指令的地址码 A 相加得到，如图 4 所示。图中 x 为变址寄存器在通用寄存器中的编号（即地址）。当计算机中还设有基址寄存器时，在计算操作数地址时还要加上基址寄存器内容。变址寄存器的内容是可以经常改变的。图中所示为处理数组的情况（x 由 $1\to m$）。

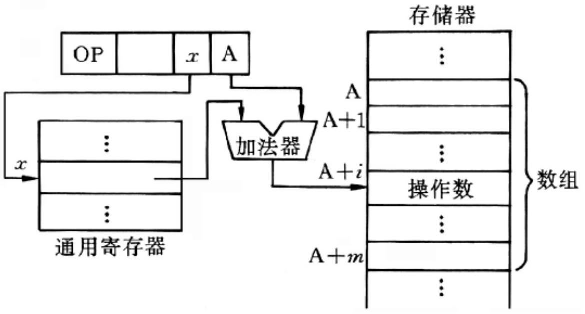


图 4　变址寻址

（6）立即数寻址　指令所需的一个操作数由指令的地址码部分直接给出，称为立即数（或直接数）寻址。这种寻址方式的特点是取指令时，同时取出一个操作数，提高了指令的执行速度，但只适用于操作数固定（即不变）的情况，且字长较短，通常用于给某一寄存器或存储器单元赋初值或置常数。

（7）相对寻址　把程序计数器 PC 的内容（即当前执行的指令的地址）和地址码部分给出的位移量相加作为操作数地址或转移指令的转移地址，称为相对寻址。转移类指令一般都用相对寻址方式，以适应浮动程序的需要。由于转移地址不是固定的，而是随着 PC 值的变化而变化，并且总是和当前指令地址（PC）相差一个固定的位移量，因此无论程序装入存储器的任何地方，均能正确运行。

有些计算机的程序和数据是分开存放的，在这种情况下，不能用相对寻址方式来确定操作数地址。

（8）隐含寻址　指令的某个操作数或操作数地址隐含在某个寄存器或指定的存储器单元中，指令不必直接给出这一操作数或操作数地址，从而可以缩短指令的长度。这种方式在微处理器中普遍采用。

（9）堆栈寻址　指令的一个操作数在栈顶，栈顶地址由一个专用的堆栈指针寄存器指出，这时指令就不必给出这一操作数或操作数地址。当取出操作数后，应修改堆栈指针，使栈顶的下一个单元成为当前栈顶。

以上这几种寻址方式可以适当组合起来使用。

对于二地址指令或三地址指令中的另一操作数和目的地址，可采用类似方法为每个地址码各自增设一个寻址方式字段。在一条指令中可同时存在多种寻址方式或采用同一种寻址方式；也可以作某些约定予以限制和简化。

假如用户用高级语言编程，则不必考虑寻址方式，因为在将高级语言程序转换成机器语言程序的编译程序中，已经考虑了寻址方式。假如用户用汇编语言编程，应对寻址方式有确切的了解，才能编出正确而又高效率的程序。

复杂指令集计算机采用了比较复杂和多样化的寻址方式，一般用高级语言编程，编译成机器语言程序后才能在计算机上运行，而**精简指令集计算机**用到的寻址方式种类则是不多的。　（王爱英）

Y

yanyu hecheng fangfa

言语合成方法(**methods of speech synthesis**)

用机械、电子、计算机等合成言语的方法。从采用的合成技术,可分为发音器官参数合成、声道模型参数合成、波形拼接合成和统计参数合成;从合成策略上,可分为频谱逼近和波形逼近。

(1) 发音器官参数语音合成　它通过直接模拟人的发音过程合成语音。这种方法定义了唇、舌、声带的相关参数。如唇开口度、舌高度、舌位置、声带张力和肺气压等。利用这些发音参数估计声道截面积函数,进而计算声波。这是对人发音过程的直接模拟,有可能产生逼真的合成语音。但由于人发音生理过程的复杂性以及理论计算与物理模拟之间的差异,合成语音的质量暂时还不理想。

(2) 声道模型参数语音合成　它通常基于一定的语音产生模型,以共振峰合成为代表。它的优点是占用的存储空间小,与语音编码相结合时数码率较低,同时合成语音具有较高可懂度,并能够较灵活地控制合成语音的音色。参数合成的主要缺点是合成语音的自然度较低。这种方法利用声道谐振特性合成语音,包括激励源模型和声道模型两部分。激励源模型包括浊激励源、清激励源和混合激励源三种,分别对应于浊音、清音以及带有较多气流的浊音情况。声道模型包括串、并联混合模型以及全并联模型。前者主要用于合成元音,后者用于合成摩擦音及爆破音等。共振峰合成器可以通过调整模型参数来改变激励源和声道参数,从而产生不同的语音。模型参数的调整基于规则,规则一般根据专家的经验确定,或者从录制的自然语音中获取。国内外已有不少基于参数合成技术的语音合成系统。较为著名的共振峰合成器是美国麻省理工学院的 D. Klatt 教授设计的串并联混合型共振峰合成器,在此基础上开发的 DEC Talk 英语文语转换系统已获得了广泛应用。

(3) 波形拼接合成　它通过对来自自然语音的语音基元(参见**汉语言语合成**)进行拼接的方式产生合成语音,具有较高的自然度。它直接把语音波形数据库中的波形拼接在一起,输出连续语流。这种语音合成技术用原始语音波形作为语音基元来替代语音参数,而且这些基元的语音波形取自自然语音的词或语句,隐含了声调、重音等细微特性,合成的语音清晰自然,质量普遍高于参数合成。这种方法主要包括基元选取合成和波形编辑合成两种。这种合成方法已经成功地应用于文语转换(TTS)系统(参见**文语转换**)中,现已有英、日、德、法、汉语等多种语言的系统问世。

基元选取合成需要一个大规模语音基元数据库。数据库中的基元从录制好的语句中切分获得。基元可以是音素、双音素、三音素、半音节、音节、词或者语句。基元可以通过语音识别中的“强制对齐”(force alignment)方法自动切分,也可以通过人工方法切分。同一个基元保存多个取自不同语句或不同位置的样本,合成时,在多个样本中,利用代价函数计算出最匹配的样本进行拼接。由于这种方法不对语音波形进行信号处理,因此可以获得最高自然度的合成语音,但是需要一个非常大的语音基元数据库支持。

不同于基元选取合成,在波形编辑合成中,每个基元只保存 1 个样本,合成时,利用基音同步覆盖叠加算法 PSOLA 对基元进行修改以获得较高自然度的合成语音。PSOLA 算法把基音周期的完整性作为保证波形及频谱平滑连续的基本前提。该算法按以下三步实施:以基音周期的整数倍为窗长,对原始波形进行分析,产生中间表示;然后对中间表示进行修改;将修改过的中间表示重新合成为语音信号。由于修改的参数不同,又分为时域 TD-PSOLA、频域 FD-PSOLA 和线性预测 LP-PSOLA。该方法较好地解决了语音拼接中的问题。

(4) 统计参数合成　又称为基于隐马尔可夫模型(hidden Markov model, HMM)的语音合成。这种方法通过对语音参数建立统计模型,进而通过最大似然准则,利用统计模型产生语音。这种方法包括训练和合成两个阶段。在训练阶段,从录制的自然语句中提取激励参数和声道参数,训练出 HMM 模型。在合成阶段,首先选取 HMM 模型的状态进行拼接,然后通过最大似然准则,从 HMM 模型的状态

参数产生激励参数和声道参数，最后，利用源-激励模型产生语音波形。这种方法结合了参数语音合成与基元选取语音合成的优点，能够获得较高自然度的合成语音，而且能够灵活调整模型参数，从而获得不同的合成语音。

参考文献

1. 蔡莲红，杨鸿武，吴志勇. 语音合成. 北京：机械工业出版社，2005

2. Zen H，Tokuda K，Black A. Statistical parametric speech synthesis. Speech Communication，2009，51：1039-1064

（蔡莲红　杨鸿武）

yanyu hechengqi

言语合成器（speech synthesizer）　能将数字代码或文本序列转换为言语信号的大规模集成电路，又称**语音合成器**。

在某些计算机应用中，将具有语音录制和回放功能的集成电路也称为语音合成器，如 UM 5100 语音处理器。这类语音合成器可以通过外接静态随机存取存储器（SRAM）作为语音记录及回放，也可通过外接可擦编程只读存储器（EPROM）或只读存储器（ROM）用作语音回放。但其基本功能仅局限于数字化录音和播音。

为了增加灵活性，言语合成器需要具备一定的自主处理能力，于是出现了内嵌中央处理器的言语合成芯片。美国 TI 公司的单片言语合成器 TSP 50C1X 和 TSP 504X 系列即属此类产品。它们采用线性预测编码（LPC）技术，以低数据率合成优质言语，因而可节省存储空间。单片言语合成器将 LPC 格形滤波器、8 位 CPU、程序 ROM 和言语数据 ROM、脉宽调制数模转换器集成在同一个芯片上，有可编程数据处理的能力，因而具有高度灵活性。

言语合成器的具体结构因所用言语处理方式不同而异，因而其结构有多种，它们的合成原理、合成内容、合成时间、比特率及声音质量等也都各有特色。言语合成方法主要有 3 种：①波形编码方式，用固定采样周期把言语信号转换为数字代码序列放入存储器中，需要时读出存储内容，并转换为模拟信号；②分析合成方式，基于分析与模拟人的发音机理，从人的言语中提取与言语参数有关的特征参数进行言语合成；③规则合成方式，以音素或音节为单位进行言语合成。

近年来，随着 TTS（Text to Speech）技术的日益成熟，许多言语合成软件中成熟的技术也下移到言语合成芯片中。图 1 给出的 XFS4041CN 就是科大讯飞公司研发的一款高性能言语合成芯片。它可以从 UART 和 SPI 接口接收文本输入，合成言语后，经过输出模块通过功放送到扬声器播出。XFS4041CN 可合成任意的中文文本，支持英文字母的合成，单次合成的文本量多达 4 KB，还支持 6 种不同的发音风格。

言语合成器在自助查询信息服务、游戏产品、教学娱乐、动态信息播报、智能仪器和家电等领域有广泛的应用前景。今后的发展方向是配置灵活、准确清晰、真人语调等。

参考文献

1. Italiano P，Ponte G，Sartori M. An integrated high quality speech synthesizer based on LPC techniques. IEEE Transactions on Consumer Electronics，1985，CE-31：501-504

2. 方建淳. 语音合成技术与单片微机综合系统. 北京：北京航空航天大学出版社，1993

（时万春　唐志敏）

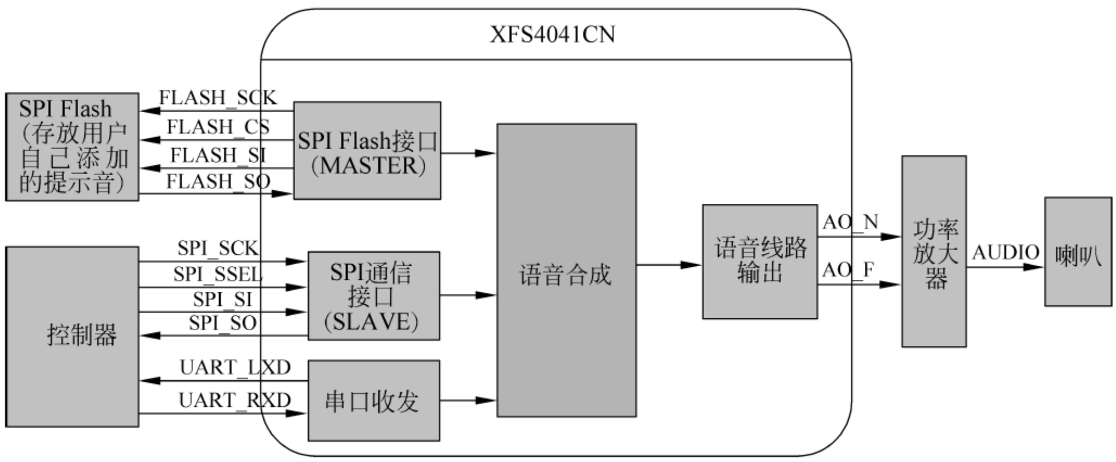


图 1　XFS4041CN 言语合成芯片系统构成框图

yanyu shibie de tezheng tiqu

言语识别的特征提取(feature extraction of speech recognition) 用信号处理技术提取语音信号中可供相互区分的特征参数。

特征提取是任何**模式识别**系统的首要工作。对言语识别而言,其基本的特征参数大致分为两类:一类反映语音信号的频谱包络特性,即声道响应特征,如共振峰参数、线性预测系数、倒谱系数等;另一类反映语音信号的时域特性,如能量、音调等。

声道响应特征通常假设在一个很小的时间帧内,声道的形状可以认为恒定不变,从而利用平稳信号处理方法可以取得瞬时的频谱特征,而语音是时变的信号,必须0.1秒以上的语音才能被感知,在这段时间内发生的时序化变化对语音的感知起关键作用。在早期,这种时序化的变化通常采用一阶差分和二阶差分来描述,差分信息的引入对识别性能提升十分明显,至今依然被广泛采用。

频谱特征中不仅含有区分说话内容的信息,也含有区分说话人特点的信息,男女童的音色有较大差异,同性别不同人之间的个体差异也千差万别,这类差异主要产生原因在于声道长度的不同,采用声道长度归一化(VTLN)技术可以很好地提高言语识别性能。深度神经网络出现后,声道长度归一化技术依然有效,但通常不需要进行声道长度归一化也能获得很好的效果。

另一种更加复杂的提取有区分意义信息的方法是从纯数学的角度去推理,完全不管其物理意义,最有代表性的是异方差线性判别分析(heteroscedastic linear discriminant analysis, HLDA)技术和特征空间的最小音子错误训练(feature-space minimum phone error training, fMPE)技术。HLDA技术以线性区分性最强为目标,将原始特征投影到低维的空间里,使得在这个空间里类间散度和类内散度比值最大化,直观地讲,类间的散度越大,则类和类之间分离得越开,说明特征的区分能力越强。fMPE训练技术以音子错误率最小化为目标,将原始特征投影到高维的后验概率空间里,再训练一个压缩变换矩阵将后验概率投影到原空间尺度上加强原特征,从而使得原始特征更具区分性。这两类技术在高斯混合模型—隐马尔可夫模型(GMM-HMM)框架下均有非常显著的性能提升,并且可以叠加使用。

近年来,随着DNN被广泛应用,以往的HLDA和fMPE等简单的线性变换技术已经被复杂的非线性变换所替代。通常认为,DNN的下层网络对特征具有正规化和再提取的作用,DNN的顶层网络具有模式分类的作用。在DNN-HMM框架下,一切特征均可回归到最原始的存在,通常以50维的梅尔滤波器组输出作为声学特征比以往的MFCC特征性能更佳,差分信息和音高曲线对改进性能依然有效。

参考文献

1. 陈永彬,王仁华. 语言信号处理. 合肥:中国科学技术大学出版社,1990

2. Stork D G, et al. Neural network lipreading system for improved speech recognition. In: Proceedings of International Joint Conference on Neural Network. 1992, 2: 289-295

(徐波 柯登峰 莫福源 方棣棠)

yanyu shibie zhong de yuyan moxing

言语识别中的语言模型(language model of speech recognition) 言语(语音)识别中,用以刻画基本语音单元与语句的数学模型。在言语识别中按语言模型利用语言学和语法知识从识别的语音中(可能有几个候选音,它们有不同的概率)挑出正确的字或词来,使语音输入的句子(语音串)变成有正确意义的文字。用语言模型可大大减少识别搜索运算量和提高识别率。

句法模型和多元文法模型(N-gram)是言语识别中最重要的两个语言模型。句法模型严格限制了句子的表达方式,超出句法描述范围的语句无法正确识别,通常用于导航、问路等限定领域。在现实生活中,许多话语并没有严格的句法和语法,而句法通常由人工总结构建,难以覆盖所有表达方式,因此句法模型不适合日常生活中的大词汇量言语识别。大词汇量言语识别通常采用多元文法模型。这种模型通过一个或多个历史词预测当前词的出现概率,按照历史的长度通常分为一元模型到五元模型,采用的历史词个数分别为零到四个,更高元的模型难以直接训练,未被广泛采纳。在训练语料足够的条件下,越高元的模型识别性能越好。与句法模型相比,多元文法模型利用回退系数对训练语料中未出现的表达方式进行概率预测,从而可以支持训练语料中没有出现的句式,一定程度上支持了用户可以不符合语法地任意说话,真正实现大词汇量的自由言语识别。

到目前为止,实用效果最好的多元文法模型训练算法依然是修改的KN折扣算法(modified Knes-

er-Ney discount)。训练时采用不同领域的语料训练成领域模型,再根据实际应用场景获得的开发集调试最佳插值系数,插值形成大语言模型,再裁剪成适当大小供言语识别使用。由于言语识别过程中需要频繁查询语言模型概率,早期的语言模型重点放在压缩空间和提高查询速度,并将一元语言模型概率附加在发音词树(lexicon tree)上用于帮助预测和剪枝。后来,研究人员把语言模型、发音词典、发音上下文、隐马尔科夫状态组合成一个统一的静态网络,该网络的每一条弧上的输入是一个马尔科夫状态,输出是一个词,弧上的概率之和等于语言模型概率。这种改进将以往的多种优化技术结合在一起,可在识别速度提高一个数量级的同时识别率还略有提升。在这个转变过程中,加权有限状态转换器(weighted finite-state transducer, WFST)的 minimize、determinize、compose 三种操作起到至关重要的作用。语言模型的优化进而转变成 WFST 的优化,出现了许多针对言语识别使用的 WFST 优化算法,包括快速而且节省内存的 compose 操作算法,最佳的 compose 执行测序,概率的向前推送(push),空弧的消除合并算法等。

随着深度神经网络在声学模型的成功应用,不少专家学者也致力于采用深度神经网络改进语言模型。采用**人工神经网络**表示语言模型的好处是历史词长度可以增加到非常大,几乎是愿意采用多少历史词都可以。有关研究表明,采用神经网络训练的多元文法模型比 ARPA 格式的多元文法模型性能更优,七元以上的语言模型性能还能进一步提升。然而神经网络计算量巨大,难以表示成有限状态机网络,严重阻碍了它在产业界的应用,通常比较适合用在词图上做语言概率重估。

参考文献

1. Waible A, Lee K F. Readings in speech recognition. San Mateo: Morgan Kaufmann Publishers Inc., 1990

2. 吴振清,郑方,等. 一种在线递增式语言模型自适应方法. 见:第 6 届全国人机语音通信学术会议,2001

(徐波 柯登峰 莫福源 方棣棠)

yanhua moxing

演化模型(evolutionary model) 一种主要针对事先不能完整定义需求,但用户可以给出待开发系统的核心需求,并且当看到其实现后,能够有效地提出反馈,以支持系统的最终设计和实现的软件开发模型。软件开发人员根据用户的需求,首先开发核心系统。当该核心系统投入运行后,用户试用之,完成他们的工作,并提出精化系统、增强系统能力的需求。软件开发人员根据用户的反馈,实施开发的迭代过程。每一迭代过程均由需求、设计、编码、测试、集成等阶段组成,为整个系统增加一个可定义的、可管理的子集。如图 1 所示。

如果在一次迭代中,有的需求不能满足用户的要求,可在下一次迭代中予以修正。

演化模型在一定程度上减少了软件开发活动的盲目性。 (王立福)

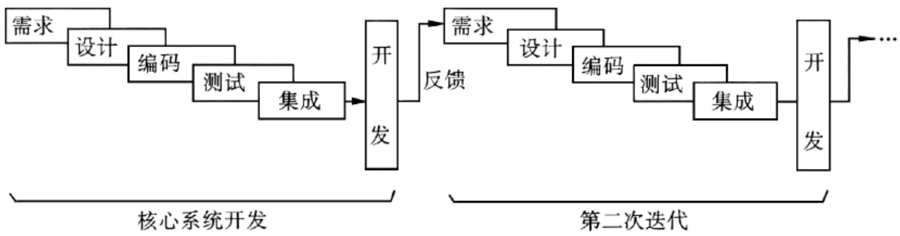


图 1 演化模型

yanyi shujuku

演绎数据库(deductive database) 具有演绎推理能力的**数据库系统**。

20 世纪 70 年代后期,人们希望通过将逻辑编程和关系数据库联合起来,建立一种具有支持强形式化表示能力,同时可以对海量数据集进行快速操作的数据处理系统。这种需要导致了演绎数据库的出现。它是**关系数据库**技术与**人工智能**技术两者结合的产物。

演绎数据库在传统数据库(主要是关系数据库)上建立一组规则和事实以及一套推理机制,使之能在现有数据库上演绎出新数据。

演绎数据库中的数据包括实数据(事实)、规则、虚数据。实数据为实际存储在数据库中的数据。

虚数据是经推理机制演绎得到的数据,它们并不实际存在于数据库中,仅在需要时才通过演绎推理得到,故称为虚数据。由于虚数据的存在,演绎数据库可获得远远多于传统数据库中的数据,但其实际占用的物理空间与传统数据库差不多。演绎数据库还具有易维护、易扩充、冗余度小和数据录入量少等优点。

演绎数据库由三部分组成:

(1) 传统数据库 由于演绎数据库建立在传统数据库之上,因此传统数据库是演绎数据库的基础。

(2) 具有对一阶谓词逻辑进行推理的演绎结构 这是演绎数据库全部功能特色所在,推理功能由此结构完成。

(3) 数据库与推理机构的接口 用于实现演绎结构与数据库在物理上的连接。

在传统数据库中,用户能检索的数据只是实际存在于关系数据库中的数据。但客观世界中的事物之间存在着多种逻辑关系,反映这些事物的数据之间同样存在着这些逻辑关系。根据已知的数据和这些逻辑关系可推出另一些在数据库中并不存在而客观又是正确的数据。

演绎数据库的推理机制是演绎数据库的核心,其主要内容是演绎推理算法,特别是递归查询算法。这些算法大体分为自顶向下和由底向上两大类。主要的自顶向下算法包括 QSQI,QSQR,QRGT 等算法;主要的由底向上的算法包括 Naive 算法与 Semi-Naive 算法。20 世纪 80 年代后期出现的 Magic Set 算法保持了自顶向下与由底向上算法的优点,同时避免了两者的缺点。此外,一些特殊的递归查询优化算法较大地提高了查询处理的效率,也受到了特别的重视,如线性递归查询优化算法、左线性递归查询优化算法、右线性递归查询优化算法及左右线性递归查询优化算法等。

演绎数据库采用一阶谓词逻辑作为表示工具。为适应数据库应用的需要,对一阶谓词逻辑进行适当改造,形成了以 Horn 逻辑为基础的 DATALOG 语言。DATALOG 语言由事实和规则两部分组成,分别用以表示外延数据库中结构与数据以及内涵数据库中的规则。演绎数据库有两种理论基础,一种是基于证明论的演绎数据库,另一种是基于模型论的演绎数据库,两种不同基础的演绎数据库在实现时采用不同的算法。比较著名的演绎数据库系统有美国斯坦福大学研制的 NAIL! 系统,美国 MCC 的 LDL 系统,墨尔本大学研制的 Aditi 系统以及美国加州大学的基于 INGRES 的演绎数据库系统 POSTGRES。演绎数据库在学术界之外还没有被广泛地使用,但是其中一些概念已经被关系数据库所采纳,用于支持 SQL 语言中的一些高级特性。

参考文献

Ullman J D. Principle of database and knowledge base systems. Vol. Ⅱ. New York: Computer Science Press, 1989

(李建中 徐洁磐 邹兆年)

yangtiao hanshu

样条函数(spline function) 一类分段(片)光滑、各段(片)交接处具有一定光滑性的函数。样条函数产生的背景是离散数据的拟合,其名称来源于船体放样时用于画光滑曲线的机械样条——弹性的细长条。高次多项式插值过程有数值不稳定的缺点,而利用分段低次多项式插值,则可在保证一定光滑性的同时取得较好的稳定性和收敛性,这种插值过程产生的函数就是(多项式)样条函数。样条函数的概念是美国数学家 I. J. Schoenberg 于 1946 年提出的,但当时没有引起人们的重视。随着科学技术和电子计算机的迅速发展以及生产部门的需要,20 世纪 60 年代中期,样条函数与计算机辅助几何设计相结合,在产品外形设计方面得到了成功应用。同时,样条函数理论也逐步完善,其应用亦逐渐扩展到各类数据的插值、拟合与平滑、数值微分与积分、微分方程的数值解法等方面。

***n* 次样条函数** 给定区间$[a,b]$上的一个分划 $\Delta: a=x_0<x_1<\cdots<x_k<x_{k+1}=b$,若函数 $S(x)$ 在区间$[a,b]$上有 $n-1$ 阶连续导数,且在每一个子区间$[x_i,x_{i+1}]$上是 n 次多项式,则称 $S(x)$ 是在$[a,b]$上关于分划 Δ 的 n 次样条函数。$x_1,x_2,\cdots,x_k$ 称为样条结点。最常用的是 $n=3$ 对应的三次样条函数,它的几何图形称为三次样条曲线。若在区间$[a,b]$上给定函数 $f(x)$,n 次样条函数 $S(x)$ 满足插值条件 $S(x_i)=f(x_i)$, $i=0,1,\cdots,k+1$,则称 $S(x)$ 为 n 次插值样条函数。$n=1$ 为分段线性插值。$n=3$ 为插值三次样条函数,它有许多最佳性质。例如,满足 $S'(a)=f'(a)$,$S'(b)=f'(b)$ 的插值三次样条唯一存在,且在一切满足上述插值条件的二次连续可微函数类中使 $\int_a^b [S''(x)]^2 \mathrm{d}x$ 取极小值。关于插值三次样条函数可参见**插值**。

插值三次样条函数有以下余项估计公式:当 $f(x)\in C^4[a,b]$时,

$$\max_{a \leqslant x \leqslant b} \left| f^{(\alpha)}(x) - S^{(\alpha)}(x) \right|$$
$$\leqslant C_{\alpha} \max_{a \leqslant x \leqslant b} \left| f^{(4)}(x) \right| h^{4-\alpha}, \quad \alpha = 0,1,2,3$$

式中,C_α 是与 $f(x)$ 无关的常数,$h = \max\limits_{0 \leqslant i \leqslant k}(x_{i+1} - x_i)$。

n 次样条函数还可表示为

$$S(x) = \alpha_0 + \alpha_1 x + \cdots + \alpha_n x^n + \sum_{i=1}^{k} \beta_i (x - x_i)_+^n / n!$$

式中记号 $u_+^p = (\max(u,0))^p$。这种表达式称为多项式样条。

n 为奇数的样条,具有与三次样条类似的多种最佳性质。而 $n=2$ 对应的二次单结点样条,即抛物线样条,则是偶次样条的代表。二次样条有许多不同于三次样条的性质。为了克服误差的传播放大,用二次样条作插值时,宜用样条结点的中点 $x_{i+\frac{1}{2}} = (x_i + x_{i+1})/2$ 作为插值结点。

分段多项式样条函数可以通过增加结点数或提高结点重数的办法来增加样条空间维数。它既有低次多项式的简单性,又有相当的光滑性和灵活适应性,同时还避免了高次多项式插值的不稳定性,因而成为函数逼近的重要工具。

多项式 B 样条 设实轴上有结点序列

$$x_{-n+1} \leqslant \cdots \leqslant a = x_0 < x_1 < \cdots < x_{k+1}$$
$$= b \leqslant x_{k+2} \leqslant \cdots \leqslant x_{k+n},$$

称 $B_{i,n}(x) = (x_{i+n} - x_i)[x_i, x_{i+1}, \cdots, x_{i+n}](t - x)_+^{n-1}$ 为 $[a,b]$ 上第 i 个(以 $x_i, x_{i+1}, \cdots, x_{i+n}$ 为结点的)n 阶($n-1$ 次)B 样条函数。各阶 B 样条函数可由如下递推公式计算:

$$B_{i,1}(x) = \begin{cases} 1, & x \in [x_i, x_{i+1}) \\ 0, & \text{其他} \end{cases}$$

$$B_{i,L}(x) = \frac{x - x_i}{x_{i+L-1} - x_i} B_{i,L-1}(x) + \frac{x_{i+L} - x}{x_{i+L} - x_{i+1}} B_{i+1,L-1}(x),$$
$$i = -n+1, -n+2, \cdots, k; L = 2,3,\cdots,n$$

不难证明,n 阶 B 样条函数具有下列性质:

(1) 局部正支撑性 即当 $x \in (x_i, x_{i+n})$ 时,$B_{i,n}(x) > 0$;否则,$B_{i,n}(x) = 0$。

(2) 叠加性 即对于任意两个结点 x_r 和 x_s,当 $x_r < x < x_s$ 时,有

$$\sum_{i=r-n+1}^{s-1} B_{i,n}(x) \equiv 1$$

(3) 易求导

$$B'_{i,n}(x) = (n-1)\left(\frac{B_{i,n-1}(x)}{x_{i+n-1} - x_i} - \frac{B_{i+1,n-1}(x)}{x_{i+n} - x_{i+1}} \right)$$

(4) 线性无关性 即 $[a, b]$ 上的 n 阶 B 样条函数 $B_{-n+1,n}(x), B_{-n+2,n}(x), \cdots, B_{k,n}(x)$ 线性无关。

区间 $[a,b]$ 上关于分划

$$\Delta: a = x_0 < x_1 < \cdots < x_{k+1} = b$$

的 $n-1$ 次多项式样条函数空间的维数是 $n+k$。由 B 样条函数的线性无关性可知,区间 $[a,b]$ 上的 n 阶 B 样条函数 $B_{-n+1,n}(x), B_{-n+2,n}(x), \cdots, B_{k,n}(x)$ 正好是该空间的一组基底。因此,区间 $[a,b]$ 上关于分划 Δ 的任一 $n-1$ 次多项式样条函数 $S_{n-1}(x)$ 可由 $\{B_{i,n}(x)\}_{i=-n+1}^{k}$ 线性表出,即

$$S_{n-1}(x) = \sum_{i=-n+1}^{k} \beta_i B_{i,n}(x)$$

在 B 样条的基础上还发展了诸如切比雪夫样条、L 样条或微分算子样条、指数样条等多种样条函数。三次样条是在小挠度的假设下得到的,而为了处理大挠度问题,人们还建立了各种非线性数学模型,发展了诸如圆弧样条和有理样条等,它们在实际应用中也很有用。

参考文献

Schmaker L L. Spline function: basic theory. New York: Wiley, 1981

(方逵 王华维)

yaogan xinxi chuli

遥感信息处理(remote sensing information processing) 从接收和记录遥感器探测到的反映地物波谱特性的原始数据,至回放出客观反映地面或对象状况的图像,所涉及的**图像复原**、几何校正、**图像变换**、**图像增强**、图像分类、统计分析、信息提取等技术和方法的统称。

遥感信息处理的目的是使图像变成便于理解和使用的形式,或提取某些特征信息供进一步分析使用。其涉及的主要技术包括空间定位、目标重建、影像匹配、**模式识别**、图像解译、数据库管理、空间信息理论、语义和非语义信息提取、**知识工程**、**人工智能**与专家系统及**计算机视觉**等,许多技术与**计算机应用技术**紧密相关,并通过**应用软件系统**实现。

遥感信息是由飞机、卫星等飞行器上的探测装置,从空间不同高度探测和收集地表物体反射或辐射的电磁波信息。各种不同的卫星和遥感平台,载有不同的传感器,探测不同的目标物,获取不同的信息,分别有陆地、海洋、气象和地球资源卫星遥感信息以及红外、微波、多光谱和高光谱遥感信息等。由遥感探测到的研究对象的特征信息通常是图像形式的遥感数据,具有时相多、内涵丰富和信息量大等特

点,例如,一颗卫星每天可获取几千兆字节的遥感数据。因此,遥感信息处理主要是针对图像形式遥感数据的处理,主要包括:

图像复原 指通过去除图像模糊、减弱噪声效应等,将图像重建成接近于或完全无退化的理想图像,使卫星遥感数据能较真实地反映地球表面特征。一般复原过程包括退化效应模型化、复原处理器设计、空域或频域上执行运算等步骤。

几何校正 对遥感图像的几何畸变进行校正处理。由于遥感器姿态角变化、扫描速度不稳定和地形起伏等因素,使图像产生几何形式或位置的失真。几何校正包括粗校正和精校正两种。前者是对图像的变形规律或遥感器在飞行时的位置和姿态进行模拟和解算,求出变换参数,再用这些参数改算所有点;而后者则通过采集地面控制点,建立校正多项式,求取各内插点的改正数,以达到校正目的。

图像变换 按一定规则将一帧遥感图像加工成另一帧图像的处理过程。变换方法有主成分分析、色度变换以及傅里叶变换等,还包括一些针对遥感图像的特定变换,如缨帽变换等。通过变换,增强所需的目标信息,使之便于识别、分析或作进一步处理。

图像增强 指应用计算机或光学设备改善图像视觉效果、并没有增加信息量的处理。增强处理的内容包括反差增强和滤波。前者常用的方法有对比度扩展、彩色增强、多谱段图像组合和变换,以改善图像上类别的判读效果;后者分为空间域滤波和频率域滤波,以提取或抑制图像的边缘、细节特征或消除噪声等。

图像分类 指将图像中所含的多个目标物通过计算机软件进行识别和区分开的遥感信息处理方法。计算机软件分类的基本原理是计算图像上每个像元的灰度特征,然后根据不同的准则进行分类。遥感图像分类又分为监督和非监督两类。监督分类需要事先确定各个类别的标准及其训练区,并计算训练区像元灰度统计特征,然后将其他像元归并到相应的类别。常用的监督分类算法有最小距离法、波谱曲线匹配法、Fisher 线性判别法等。非监督是一种未知类别标准的分类方法,它直接根据像元灰度特征之间的相似性和相异性进行合并与区分,形成不同的类别,常用的非监督分类算法有聚类法和分裂法等。近年来,专家系统分类法和神经网络分类法已在遥感图像分类中获得较好的分类效果。

图像融合 指将多种遥感平台、多时相遥感数据之间以及遥感数据与非遥感数据之间的信息组合匹配技术。不同尺度遥感影像间的自动配准仍然主要基于点和线特征,在点匹配方向目前主要是寻找最优化搜索策略,而线特征匹配是当前的研究热点。在融合方法方面,除改进传统算法外,已出现许多新的融合方法,如小波变换、多尺度变换和智能变换等,这种多源遥感数据融合技术提高了遥感信息的可应用性,更有利于地学综合分析。

遥感信息处理分为预处理和应用处理两部分。预处理技术主要针对遥感数据获取的特色,进行一系列的校正,以消除引入的误差和干扰等,使遥感数据能较真实地反映它所代表的物体结构、光谱等方面的信息。应用处理技术则从研究对象的地学、生物学或环境科学等方面的综合知识入手,研究其波谱特性及其在探测和处理过程中的变化,结合遥感基础理论和基础知识,并通过图像融合技术,使得能从遥感记录中获得研究对象的丰富内容,提取到能反映研究对象本质特征的专题信息。如何才能有效地提取这种专题信息,是一个颇具潜力的研究方向。

参考文献

1. 陈述彭. 遥感大辞典. 北京: 科学出版社,1990

2. 中国测绘学会. 测绘科学与技术学科发展报告(2009—2010). 北京: 中国科学技术出版社,2010

(黄杏元)

yaogan xinxi chuli xitong

遥感信息处理系统(remote sensing information processing system) 面向遥感信息处理的**应用软件系统**。遥感信息处理是从接收和记录遥感器探测到的反映地物波谱特性的原始数据,至回放出客观反映地面或对象状况的图像,所涉及的图像复原、几何校正、图像变换、图像增强、图像分类、统计分析、信息提取等技术和方法的统称。

遥感信息处理系统的目的是使图像变成便于理解和使用的形式,或提取某些特征信息供进一步分析使用。其涉及的主要技术包括空间定位、目标重建、影像匹配、**模式识别**、图像解译、**数据库**管理、空间信息理论、语义和非语义信息提取、知识工程、**人工智能与专家系统**及计算机视觉等,许多技术与计算机应用技术紧密相关。

遥感信息是由飞机、卫星等飞行器上的探测装置,从空间不同高度探测和收集地表物体反射或辐射的电磁波信息。各种不同的卫星和遥感平台,载

有不同的传感器,探测不同的目标物,获取不同的信息,分别有陆地、海洋、气象和地球资源卫星遥感信息以及红外、微波、多光谱和高光谱遥感信息等。由遥感探测到的研究对象的特征信息通常是图像形式的遥感数据,具有时相多、内涵丰富和信息量大等特点,例如,一颗卫星每天可获取几千兆字节的遥感数据。因此,遥感信息处理主要是针对图像形式遥感数据的处理,主要包括:

图像复原　指通过去除图像模糊、减弱噪声效应等,将图像重建成接近于或完全无退化的理想图像,使卫星遥感数据能较真实地反映地球表面特征。一般复原过程包括退化效应模型化、复原处理器设计、空域或频域上执行运算等步骤。

几何校正　对遥感图像的几何畸变进行校正处理。由于遥感器姿态角变化、扫描速度不稳定和地形起伏等因素,使图像产生几何形式或位置的失真。几何校正包括粗校正和精校正两种。前者是对图像的变形规律或遥感器在飞行时的位置和姿态进行模拟和解算,求出变换参数,再用这些参数改算所有点;而后者则通过采集地面控制点,建立校正多项式,求取各内插点的改正数,以达到校正目的。

图像变换　按一定规则将一帧遥感图像加工成另一帧图像的处理过程。变换方法有主成分分析、色度变换以及傅里叶变换等,还包括一些针对遥感图像的特定变换,如缨帽变换等。通过变换,增强所需的目标信息,使之便于识别、分析或做进一步处理。

图像增强　指应用计算机或光学设备改善图像视觉效果,并没有增加信息量的处理。增强处理的内容包括反差增强和滤波。前者常用的方法有对比度扩展、彩色增强、多谱段图像组合和变换,以改善图像上类别的判读效果;后者分为空间域滤波和频率域滤波,以提取或抑制图像的边缘、细节特征或消除噪声等。

图像分类　指将图像中所含的多个目标物通过计算机软件进行识别和区分开的遥感信息处理方法。计算机软件分类的基本原理是计算图像上每个像元的灰度特征,然后根据不同的准则进行分类。遥感图像分类又分为监督和非监督两类。监督分类需要事先确定各个类别的标准及其训练区,并计算训练区像元灰度统计特征,然后将其他像元归并到相应的类别;常用的监督分类算法有最小距离法、波谱曲线匹配法、Fisher 线性判别法等。非监督是一种未知类别标准的分类方法,它直接根据像元灰度特征之间的相似性和相异性进行合并与区分,形成不同的类别,常用的非监督分类算法有聚类法和分裂法等。近年来,专家系统分类法和神经网络分类法已在遥感图像分类中获得较好的分类效果。

图像融合　指将多种遥感平台、多时相遥感数据之间以及遥感数据与非遥感数据之间的信息组合匹配技术。不同尺度遥感影像间的自动配准仍然主要基于点和线特征,在点匹配方向目前主要是寻找最优化搜索策略,而线特征匹配是当前的研究热点。在融合方法方面,除改进传统算法外,已出现许多新的融合方法,如小波变换、多尺度变换和智能变换等,这种多源遥感数据融合技术提高了遥感信息的可应用性,更有利于地学综合分析。

遥感信息处理分为预处理和应用处理两部分。预处理技术主要针对遥感数据获取的特色,进行一系列的校正,以消除引入的误差和干扰等,使遥感数据能较真实地反映它所代表的物体结构、光谱等方面的信息。应用处理技术则从研究对象的地学、生物学或环境科学等方面的综合知识入手,研究其波谱特性及其在探测和处理过程中的变化,结合遥感基础理论和基础知识,并通过图像融合技术,使得能从遥感记录中获得研究对象的丰富内容,提取到能反映研究对象本质特征的专题信息。如何才能有效地提取这种专题信息,是一个颇具潜力的研究方向。

参考文献

1. 陈述彭. 遥感大辞典. 北京:科学出版社,1990

2. 中国科学技术协会主编,中国测绘学会编著. 测绘科学与技术学科发展报告(2009—2010). 北京:中国科学技术出版社,2010

(黄杏元)

yewu jiedian jiekou

业务节点接口(service node interface, SNI)

接入网(AN)与业务节点(SN)之间的接口,它独立于业务节点和交换机,支持宽泛的接入类型,把不同业务的 SN 通过不同的 SNI 与 AN 相连,进而向用户提供多种不同的业务服务。AN 有多种接入类型,相应 SN 也有多种,对于不同的接入类型和不同的 SN,有不同的 SNI 与之对应。

SN 根据其支持的接入类型和接入承载能力,可分为三种:仅支持一种专用接入类型;可支持多种接入类型,但所有接入类型的接入承载能力相同;可支持多种接入类型,但每种接入类型的接入承载能力不同。

业务节点(SN)不同于传统的网络节点(NN)。NN 只是描述电信网中交换节点的交换功能,而 SN 不仅描述了其交换功能,而且描述了其所交换的业务和种类。SN 可能有多种类型,包括特定业务的业务节点和模块化业务节点。

参考文献

1. 通信行业标准:接入网设备测试方法 第二代甚高速数字用户线(VDSL2). YD/T 2278—2011. 2011

2. 通信行业标准:数据通信名词术语. YD/T 1133—2001. 2001

(程时端　马严)

yewu zhineng

业务智能(business intelligence, BI)　综合运用**数据仓库**、**联机分析处理**、**数据挖掘**和**大规模数据可视化**等技术,从大量数据中获取辅助决策知识的一种工具软件与解决方案。BI 通常服务于企业业务经营与管理的智能决策,故又称商业智能或商务智能。

BI 的概念出现于 1989 年,初衷是整合各类数据和统计报表,进而运用数据管理、数据仓库(或数据集市)、数据分析、数据挖掘等技术和工具,从企业信息系统中出现的看似分散而杂乱的大量数据中,找出有一定规律的知识,以支持企业管理决策。2006—2007 年开始出现第二代业务智能 BI 2.0,它能够在某些突发事件发生时或发生之前,为用户提供更主动的辅助决策反应。目前,典型的业务智能系统应用于客户关系分析、反洗钱、反诈骗、市场细分、信用评估、产品收益、库存运作以及与商业风险相关的领域。

与**应用软件系统**的逻辑结构类似,BI 的技术架构也有三个层次:(1)**数据仓库**　相当于资源管理层,旨在提供 BI 数据分析用的原始数据资源保障。数据仓库是为**决策支持系统**提供基础数据的分析型数据库,具有数据量巨大、面向主题、集成的、数据不再修改和随时间增加等特点,其构建过程是根据预先设计好的逻辑模式从分布在企业内部各处的联机事务处理数据库中提取数据,经必要的转换和装载,最终合并成全企业统一的全局视图的过程。(2)**联机分析处理**和**数据挖掘**　相当于业务逻辑层,是 BI 进行数据分析的主体,经选择适当的数据分析技术和工具,数据和信息将变为辅助决策的知识。其中,**联机分析处理**工具支持人们从多种角度(例如时间,地区,商品种类,及相应的销售额等)快速灵活地对数据仓库中的海量数据进行联机的复杂查询和多维分析处理,通过切片、切块、旋转、向上综合和向下钻取等多维分析操作,使用户能从多个角度多个层次观察和剖析数据,从而获得对数据内涵的深入了解;数据挖掘工具是一种基于数据库、人工智能和数理统计等方法,对大量看似无序的数据,通过关联分析、分类分析、聚类分析、演变分析和异常分析等技术,从中挖掘出有规律的预测和决策知识。(3)**信息可视化**　相当于应用软件系统的应用表现层,通过图形图像处理等技术,对知识进行可视化处理,为 BI 的大规模非数值型信息资源提供报表、图形等视觉呈现,以帮助人们对知识的理解。与科学计算可视化不同,信息可视化侧重于抽象数据集,如非结构化文本或者高维空间中的数据。

参考文献

1. Chaudhuri S, Dayal U, Narasayya V. An overview of business intelligence technology. CACM, 2011, 54(8)

2. Wikipedia. Business intelligence. http://en.wikipedia.org/wiki/Business_intelligence

(刘江宁　吴泉源)

yijie luoji

一阶逻辑(first order logic)　研究由个体、函数及关系构成的命题,由这些命题经使用量词和命题联结词构成的更复杂的命题以及各种命题之间的推理关系的逻辑。在一阶逻辑中,量词仅作用于个体变元。一阶逻辑是数理逻辑中发展得最为成熟的部分。在为数学的语言和推理建立形式系统的过程中,它处于核心地位,又称谓词逻辑。

传统逻辑主要是指古代希腊的亚里士多德逻辑,在中世纪被认为是金科玉律,完美无缺,不容许有任何更改。但是到了 19 世纪,人们觉得它有很多缺点,需要改革。传统逻辑仅限于讨论主宾式语句和三段论形式的推理,并且缺乏对于量词的研究。A. De Morgan 研究和发展了关系逻辑,提出了论域的概念。W. Hamilton 对量词进行了研究。G. Frege 于 1897 年建立了第一个谓词逻辑的形式系统。B. Russell 和 A. Whitehead 于 1910 年在他们的数学名著《数学原理》中总结了前一段的成果,建立了一个完全的谓词演算。1930 年 K. Godel 证明了谓词演算的完全性。

在命题逻辑中,不进一步分析原子命题的内部结构,原子命题被看作是不可分割的最小单位,因而

不能包括某些正确的推理。例如以下的推理：

每个有理数都是实数，

1 是有理数，

所以，1 是实数。

在这个推理中，前提和结论是三个不同的命题。在命题逻辑中，其推理形式只能表示成

$$\frac{\begin{array}{l}p\\q\end{array}}{r}$$

这不是命题逻辑中正确的推理形式。但这个推理是正确的，其正确性只有通过分析各命题中的个体词、谓词、量词才能揭示出来。这正是一阶逻辑所研究的。

在一阶逻辑中描述一个数学理论，首先会涉及这个理论所讨论的对象、定义在这些对象上的函数以及这些对象之间的关系。数学理论所讨论的对象称为个体，由个体组成的非空集合称为论域或个体域。相等关系是经常需要用到的，在一阶逻辑中用一个特殊的符号，即等号“=”表示它。

为了表达每个个体都有某性质，在一阶逻辑中引进了全称量词∀。为了表达至少有一个个体有某性质，在一阶逻辑中引进了存在量词∃。例如，设论域是整数集，$\mathbf{N}(x)$表示x是自然数。$\forall x\,\mathbf{N}(x)$表示命题“每个整数都是自然数”，这是一个假命题。$\exists x\,\mathbf{N}(x)$表示命题“至少有一个整数是自然数”，这是一个真命题。

一阶逻辑使用的形式语言称为一阶语言，它的符号包括以下几类。

（1）个体变元$x,y,z,\cdots$，简称变元。

（2）函数符号$f,g,h,\cdots$；个体常元$a,b,c,\cdots$，谓词符号$P,Q,R,\cdots$，其中包括二元谓词符号“=”。

（3）命题联结词$\neg,\to$和全称量词∀。

通常称函数符号、个体常元、除等号“=”之外的谓词符号为非逻辑符号，而称其余符号为逻辑符号。不同的一阶语言有不同的非逻辑符号，而所有一阶语言的逻辑符号都是相同的。

因为$\wedge,\vee,\leftrightarrow$可用$\neg,\to$定义，所以可取$\neg,\to$为基本联结词（参见**命题逻辑**）。因为存在量词∃可用∀和$\neg$定义，$\exists x\,\mathbf{N}(x)$与$\neg\forall x\neg\mathbf{N}(x)$表达的命题之真假意义相同，所以仅取∀为基本量词。

项的形成规则如下：

（1）变元和个体常元是项；

（2）若f是n元函数符号，$t_1,t_2,\cdots,t_n$是项，则$f(t_1,t_2,\cdots,t_n)$是项；

（3）每个项都可以通过有穷次应用（1）和（2）获得。

公式的形成规则如下：

（1）若P是n元谓词符号，$t_1,t_2,\cdots,t_n$是项，则$P(t_1,t_2,\cdots,t_n)$是公式，也称为原子公式；

（2）若A是公式，则$\neg A$是公式；

（3）若A,B是公式，则$(A\to B)$是公式；

（4）若A是公式，x是变元，则$\forall x\,A$是公式；

（5）每个公式都可以通过有穷次应用（1）～（4）获得。

如果变元x出现在公式A中形如$\forall x\,B$的部分，则称x在A中的这次出现为约束出现，否则称为自由出现。例如，在公式$P(x)\to\forall x\,Q(x)$中，第一个x是自由出现，后两个x是约束出现。如果变元x在公式A中有自由出现，则称x为A的自由变元。没有自由变元的公式称为闭公式。常用$A_x[t]$表示将公式A中x的所有自由出现代之以项t所得到的公式。如果$A_x[t]$和A中的变元的约束出现数相同，则称t对A中的x是可代入的。例如$f(z)$对于$P(x)\to\forall y\,Q(x,y)$中的x是可代入的，而$f(y)$对于$P(x)\to\forall y\,Q(x,y)$中的x不是可代入的。

设L是一个一阶语言。可指定一个非空集合为论域，将等号“=”解释为论域上的相等关系，为L的每个个体常元指定论域中的一个个体，为L的每个n元函数符号指定论域上的一个n元函数，为L的每个非逻辑的n元谓词符号指定论域上的一个n元关系，就得到L的一个结构。闭公式在一个结构中的解释是一个命题。例如，指定论域为正整数集，个体常元a解释为2，$P(x)$解释为x是素数，$L(x,y)$解释为$x\leqslant y$，则闭公式$P(a)\wedge\forall y(P(y)\to L(a,y))$就解释为真命题：2是最小的素数。有自由变元的公式在一个结构中的解释还不是一个命题，因为自由变元值不确定。例如，在上面所给的结构中，如果给变元z赋值2，则公式$P(z)\wedge\forall y(P(y)\to L(z,y))$解释为真命题；如果给变元$z$赋值3，则该公式解释为假命题。设给定一个结构$U$，如果给自由变元赋予论域中任何个体，公式$A$都被解释为真命题，则称$A$在$U$中有效，记为$U\models A$。如果公式$A$在每个结构中有效，就称$A$为逻辑有效式或永真式，记为$\models A$。例如，公式$\forall xP(x)\to P(y)$是逻辑有效式，而$P(y)\to\forall xP(x)$不是逻辑有效式。

谓词演算把逻辑有效式组成了一个完全形式化的公理系统。在谓词演算中，取某些逻辑有效式为公理，并规定了一些推理规则，以推导出所有的逻辑

有效式。人们给出了许多等价的谓词演算系统。下面举出其中的一个。

取以下 7 种形式的公式为公理：

$A \to (B \to A)$

$(A \to (B \to C)) \to ((A \to B) \to (A \to C))$

$(\neg B \to \neg A) \to (A \to B)$

$x = x$

$x = y \to (A \to A_x[y])$，其中 y 对于 A 中的 x 可代入

$\forall x\, A \to A_x[t]$，其中项 t 对于 A 中的 x 可代入

$\forall x (A \to B) \to (A \to \forall x\, B)$，其中 x 不是 A 的自由变元

推理规则有以下两条：

分离规则　由前提 A 和 $A \to B$ 推出结论 B。

概括规则　由前提 A 推出结论 $\forall x\, A$。

谓词演算的定理是这样定义的：

(1) 每个公理都是定理；

(2) 如果 A 和 $A \to B$ 是定理，则 B 是定理；

(3) 如果 A 是定理，则 $\forall x\, A$ 是定理；

(4) 每个定理都可以通过有穷次应用(1) ~ (3)获得。

若公式 A 是定理，则记为 $\vdash A$。

谓词演算的公理都是逻辑有效式。推理规则保证，当前提都在某结构中有效时，结论也在该结构中有效。因此，谓词演算的定理都是逻辑有效式。这个性质称为谓词演算的可靠性。反之，每个逻辑有效式都是谓词演算的定理。这是谓词演算的完全性，由 K. Godel 于 1930 年证明。

公式是不是逻辑有效式是半可判定的，不是可判定的(参见**可计算性理论**)。

可以用一阶逻辑刻画数学理论，常把谓词演算的公理称为逻辑公理。除了逻辑公理之外，数学理论还需要非逻辑公理，它们刻画了该数学理论的研究对象的共同性质。这样的数学理论称为一阶理论。如果一阶理论 T 的每个非逻辑公理都在结构 U 中有效，就称 U 是 T 的模型。下面举出几个常见的一阶理论。

全序理论只有唯一的非逻辑符号，即二元谓词符号≤。它有 4 个非逻辑公理：

$x \leq x$

$(x \leq y \wedge y \leq x) \to x = y$

$(x \leq y \wedge y \leq z) \to x \leq z$

$x \leq y \vee y \leq x$

每个全序结构都是它的模型。

群论有两个非逻辑符号：个体常元 e 和二元函数符号“·”。它有 3 个非逻辑公理：

$\forall x \forall y \forall z ((x \cdot y) \cdot z = x \cdot (y \cdot z))$

$\forall x (x \cdot e = x \wedge e \cdot x = x)$

$\forall x \exists y (x \cdot y = e \wedge y \cdot x = e)$

每个群都是它的模型。

初等数论的非逻辑符号有：个体常元 0，一元函数符号 S，两个二元函数符号“+”和“·”。它的非逻辑公理为：

$\forall x \neg (S(x) = 0)$

$\forall x \forall y (S(x) = S(y) \to x = y)$

$\forall x (x + 0 = x)$

$\forall x \forall y (x + S(y) = S(x + y))$

$\forall x (x \cdot 0 = 0)$

$\forall x \forall y (x \cdot S(y) = (x \cdot y) + x)$

$A_x[0] \wedge (\forall x (A \to A_x[S(x)])) \to \forall x A$

自然数系统是它的一个模型。

一阶逻辑有很强的表达能力，用一阶理论能够刻画许多数学结构。但是，一阶逻辑不是万能的，其表达能力受到一定的限制。

如果一阶理论 T 的模型的类恰好是结构的类 S，就称 T 刻画 S。例如，全序理论刻画全序结构的类，群论刻画群的类。常把论域是有穷集的结构称为有穷结构。

一阶逻辑有可靠且完全的公理系统，有许多良好的性质，是数理逻辑中发展得最为成熟的部分。在计算机科学和人工智能的各个领域，如数据结构理论、程序设计语言的形式语义，程序正确性证明、软件规范、程序变换、逻辑程序设计、定理的机器证明、常识表示和推理等方面，都广泛地应用了一阶逻辑的概念、理论和方法。

参考文献

1. 王宪钧. 数理逻辑引论. 北京：北京大学出版社，1982

2. 王浩. 数理逻辑通俗讲话. 北京：科学出版社，1981

3. Kleene S C. 元数学导论. 莫绍揆，译. 北京：科学出版社，1984

(何自强)

yiliao xinxi xitong

医疗信息系统(healthcare information system)　医疗卫生领域专用的**应用软件系统**。

医疗信息系统始于 20 世纪 60 年代中期，主要是利用**计算机软件**做一些病理分析和辅助诊断的工

作。80年代涌现了大批医疗诊断**专家系统**和医院**管理信息系统**。90年代,随着**数字图像处理**等技术和数字医疗设备的发展,数字医疗系统成为医院的重要装备,基于**互联网**的远程诊疗系统也取得了重要进展。进入21世纪以来,一体化医院信息系统和医疗卫生信息综合服务系统成为现代医疗信息系统的主流。

医疗信息系统大致可以分为三类:

(1)数字医疗系统　指基于核磁共振、CT、B超、心电图、脑电图、电子肠镜、纤维胃镜和伽马刀等数字医疗设备,协助医务人员进行病理分析和诊疗的信息系统。一般由嵌入在数字医疗设备中的嵌入式应用软件系统、后端的影像资料库与医疗信息分析系统、前端的二维或三维显示系统与人机交互系统组成。

(2)医院信息系统　包括基于电子病历和诊疗知识库并为病人诊疗提供辅助决策信息的临床信息系统,提供医院业务管理和人财物管理的医院行政管理信息系统,以及辅助医院管理人员实施决策和监管调控的医院资源规划系统等。

(3)医疗卫生信息服务系统　指服务于广大城乡居民,为他们提供有关医疗、保健与公共卫生管理等信息综合服务的应用软件系统,包括医疗保障信息系统,居民健康档案管理信息系统,公共卫生信息服务系统以及网上预约、网上挂号等医院便民服务系统等。

医疗信息系统的发展趋势主要表现在:①集成化　通过各类信息资源的共享,实现医院内部、医院之间以及医院、病人与卫生管理部门之间相关信息系统的无缝集成。②泛在化　基于电子病历,使每个人在网上都有一份完整的个人电子健康记录,基于物联网技术,主管医师或护士在任何时间任何地点都能够通过移动无线网等网络设施感知其病人的健康信息和当前的健康状况,卫生部门则可以随时了解到医药卫生事业运行动态。③智能化　充分运用**知识工程**和智能技术,在发展数字医疗系统的同时,不断提高临床诊疗业务和医院管理的信息化水平,数字医院将得到快速发展。(吴泉源　任连仲)

yidong IP

移动 IP(mobile IP)　为满足移动节点在移动中保持其连接性而设计的网络技术,实现跨越不同IP网段的漫游功能。移动IP的正式名称为IP移动性支持(IP mobility support)。

移动IP技术由互联网工程任务组IETF提出,主要为克服传统的IP寻址模式在移动性方面的限制。传统的IP寻址模式主要为固定环境设计和优化,节点的IP地址包含了用于说明所属子网的网络前缀。当节点移动到一个新子网时,尽管可以通过改变节点IP地址或通过传播特定路由实现网络访问功能,然而这两种方式均存在断开已有的传输层连接问题以及规模扩展性问题。因此出现了移动IP技术。

移动IPv4通过引入家乡代理和外部代理进行移动性管理,主要功能包括代理发现、注册和隧道路由。家乡代理和外部代理分别在家乡网络和外地网络发送消息,向移动节点通知自己的存在。移动节点据此判断是处于家乡网络还是外地网络。在家乡网络时,移动节点进行正常的IP通信。进入外地网络后,移动节点获得转交地址,之后向家乡代理发送注册消息,告知转交地址。发送给移动节点的数据包通过正常的IP路由到达移动节点的家乡网络。家乡代理截获这些数据包,重新封装,添加新的目的地址为转交地址、源地址为家乡代理地址的IP报头,通过隧道转发到移动节点的当前位置,最后恢复为封装前的IP报文。移动IPv6取消了外部代理的概念,当移动到一个外部网络时,移动节点使用IPv6地址自动配置功能获得转交地址,然后向家乡代理完成注册操作。

移动IP相关的协议规范主要有两个,分别为移动IPv4(RFC 5944)和移动IPv6(RFC 3775)。

参考文献

1. Perkins C, et al. RFC 5944: IP mobility support for IPv4, Revised. 2010

2. Johnson D, et al. RFC 3775: IP mobility support in IPv6. 2004

(苏金树)

yidong cunchuqi

移动存储器(mobile storage device)　便携式数据存储装置的总称。移动存储器包括移动硬盘、移动式**固态盘**(SSD)、U盘和各种规格的记忆卡等。它们是便携式独立保存信息的完整装置,并可方便地通过计算机或数码相机等信息设备配备的标准外置接口(如USB接口)读写信息,主要用于信息的交换、保存和携带。内置式硬盘驱动器和光盘驱动器不具备移动性,而单独的光碟介质和软盘介质需要通过驱动器而不是直接通过标准的外置式接口读取,因而都不能称为移动存储器。

移动硬盘一般指具有 USB 接口(参见**外存储设备接口**)的可携带式硬盘。在很多情况下,它是将普通内置硬盘装在一个具有接口转换功能盒子中而构成。它具有容量大、速度快的特点,用于备份和携带大量的文件。

移动式固态盘外部具有 USB 接口,内部由闪存芯片构成的固态盘组成。移动式固态盘比移动硬盘速度更快,并具有抗震动、冲击和耐受高低温的优点,但价格较高。

U 盘是具有 USB 接口的**快闪存储器**,它完全取代了过去的软盘,是应用最为广泛的保存、携带和交换信息的装置。

记忆卡也是常用的移动存储器,常用于数码相机、摄像机和手机的信息存储。记忆卡有多种规格,常用的有 CF(compact flash)、SD(secure digital)、MS(memory stick)、MMC(multi media card)等规格和它们的变种 MINI-SD、RS-MMC、MS PRO、MS Duo 以及专用于手机的超小型记忆卡 T-Flash。　(谢长生)

yidong jiqiren

移动机器人(mobile robot)　指在工作环境中可移动的自主机器人。与工业机器人不同,它不固定在某一位置,具有更大的工作空间和灵活性。最早的移动机器人出现在 20 世纪 60 年代,是美国斯坦福大学研制的世界上第一台移动机器人 Shakey。随后,许多大学和科研机构开展了移动机器人方面的研究。

移动机器人学是一个新兴的领域,它的基础包括许多工程和科学学科,从机械、电子、自动化到**人工智能**、计算机和社会科学。因其应用和移动方式不同,研究内容有很大的差别。其基本内容包括传感器、移动机构、即时定位与地图构建(SLAM)、路径规划、控制理论与导航方法等。

移动机器人作为一类自主系统,其最基本的能力之一是能够感知自身状态和外部环境,而这是通过不同的传感器来实现的。这些传感器的种类非常广泛,分为内、外传感器,包括各种光、声、电和磁物理参量的视觉、听觉、触觉、力觉传感器,以及惯性、激光测距传感器、编码器和**全球定位系统**(GPS)等。

移动机器人的行走机构,使得机器人在工作环境中能够无约束地运动。主要有轮式、履带式、足式和复式机构等。其中轮式运动应用十分广泛,但其穿越能力不如履带式的移动机器人,足式结构的移动机器人则具有更强的地形适应能力,而复式结构综合了轮式结构的高速机动性和履带式穿越能力强或者足式机构的地形适应能力强的优点。

即时定位与地图构建(SLAM)指的是移动机器人在未知环境运动过程中,基于位置估计和传感数据进行自身定位和同步地图创建,是自主导航研究的一个关键问题。定位不仅仅确定移动机器人在地球参考坐标系中的绝对位置,相对于目标的位置也同样重要,甚至需要建立一个环境模型,即一幅地图,以规划一条达到目标的路径。

路径规划利用环境感知信息,按照某优化指标,在机器人运动起始点和目标点之间确定一条可行的、无碰撞的安全路径,可分为基于地图的全局路径规划和基于传感器信息的局部路径规划。常用的规划方法有势场法、空间拓扑法和 D^* 算法等。

移动机器人的控制分为遥控和自主导航两种方式。遥控利用无线、Internet 和蓝牙等通信手段,传输反馈和控制信息,实现移动机器人的远程操控。实时性和安全性仍是移动机器人遥控有待解决的问题。自主导航控制则完全利用内、外传感器信息,完成移动机器人定位,进行自主判断和决策,进而实现无碰撞的导航控制。

移动机器人有着非常广泛的应用前景,能够代替人从事危险、恶劣环境下的作业,比如星空和海洋探测任务,甚至还能够打理家务。按照工作环境分为室内和室外两类移动机器人。室内移动机器人有家庭服务机器人和导游机器人等,一般在已知的、结构化的室内环境下工作。室外移动机器人可分为智能车、空中无人机(UAV)、水下无人机器人(AUV)和极地机器人等,它们工作在未知的半结构化和非机构化外部环境。

参考文献

1. 蔡自兴,等. 未知环境中移动机器人导航控制理论与方法. 北京:科学出版社,2009

2. Siegwart R, Nourbakhsh I R. Introduction to autonomous mobile robots. MIT Press, 2004

(杨唐文　韩建达)

yidong jisuan

移动计算(mobile computing)　利用可移动的计算设备,通过无线网络交换信息,为用户提供服务的一种计算。移动计算是利用移动硬件提供的设备支撑、移动通信提供的网络环境,基于移动软件来完成的一种分布式计算。

移动硬件:移动硬件可以分为移动设备以及构

成这些设备的移动组件。其中,移动设备包括**平板计算机**、智能手机、电子书阅读器、可穿戴设备、智能路由器、智能车等;**核心组件**包括为移动设备提供功能支撑的各个组件,比如**地理信息系统**(GPS),Wi-Fi 网卡,电池,显示屏,摄像头,麦克风。随着移动组件性能的提升,体积的减小,以及价格的下降,移动设备的功能愈加强大,同时便携性和易用性提高,正朝着可穿戴,高智能的方向(比如 Google Glass,Siri)发展。

移动通信:移动通信指的是移动设备通过接入的无线网络进行语音和数据通信的技术。根据环境的不同,不同的无线网络(参见**无线网络技术**)为移动通信发挥着作用。移动通信技术主要解决的问题包括:带宽(为移动设备提供高速的数据访问),能耗(降低通信给移动设备带来的能耗),安全(为无线网络这种开放的网络接入提供可靠的安全机制)等。

移动软件:移动软件基于移动硬件和移动通信技术,为用户提供移动环境下的各种服务。移动软件包括处理用户交互的**人机交互技术**(比如触摸、语音、图像、动作等的识别),社交网络、电子商务(比如手机支付)等移动互连网络应用技术,个人隐私等管理和协助软件。移动硬件性能的提高、传感器种类的丰富、**人工智能**技术的进步为移动软件的发展提供了广阔的空间。

移动计算是分布式系统、无线网络、嵌入式技术、人机交互技术、人工智能等领域的交叉和融合。上述领域的发展推动着移动计算技术的发展,进而扩大了移动计算的用户规模,丰富了移动计算的应用场景,给移动计算带来了新的需求和挑战。

参考文献

1. 徐明,曹建农,彭伟. 移动计算技术,北京:清华大学出版社,2008

2. ACM Mobility Tech Pack Committee. ACM Tech Pack on Mobility,December,2011

(周悦芝)

yidongshi jisuanji

移动式计算机(mobile computer) 在移动过程中可以进行信息处理的、可随身携带的、体积很小的**微型计算机**,又曾称为便携式计算机。移动式计算机要具备三个方面的能力:移动通信、移动硬件和移动软件。移动通信要解决基础设施与网络终端的通信问题,包括其性能、协议、数据格式和具体实现技术;移动硬件要采用适合于移动信息处理的器件和元件;移动软件要处理移动应用中的特点和需求。

移动式计算机可分为**笔记本计算机**、手持计算机和可穿戴计算机(参见**可穿戴计算**)等。这些移动式计算机都能通过无线或有线传输方式与通信网络连接以传输和处理信息,它们具有成为因特网或其他通信网络终端设备的功能,有的还具有电话和传真等功能,如**个人数字助理**(PDA)、全球定位系统(GPS)终端、现代手机,尤其是最近发展非常迅速的无线终端等。它们都具有能处理业务信息、个人信息、上网浏览和进行电子邮件通信的功能。另外,它们还具有节能的共同特点,即采用节能的低电源电压的、互补金属氧化物半导体(CMOS)工艺的芯片以及节能的外围设备,如代替硬磁盘存储器的闪存、液晶显示屏。在输入方面,常采用笔输入技术,即使用模式识别技术,以笔输入方式代替键盘输入。有时还需要语音输入。移动式计算机发展很快,为了便于大规模生产,建立了很多标准。例如,为了使移动式计算机与电话结合,建立了电话应用编程接口(TAPI);为了使移动式计算机板卡的可互换性更高,建立了可互换板卡体系结构(ExCA);特别是为了方便地扩展存储容量和外围设备,采用了由个人计算机存储卡国际协会(PCMCIA)所制定的标准。移动式计算机可选择的电路板卡种类繁多,例如存储卡、调制解调卡、以太网卡、传真卡等。移动式计算机的外围设备也日趋多样化,如跟踪球、触摸屏、控制杆等。为了使移动式计算机有硬备份或复制件输出,也出现了一些小型便携式输出设备,如热转印印刷机(参见**非击打式印刷机**)等。移动式计算机的操作系统一般采用嵌入式操作系统。嵌入式操作系统的特点是其代码十分紧凑,需要的存储总容量较小,以节省电源消耗量。嵌入式操作系统往往具有基本功能的核,然后根据应用的不同或系列产品的升级,再在核外面扩充操作系统功能。很多嵌入式操作系统是针对专门用途而开发的,有的还要求有实时的功能。移动式计算机中处理机的功能范围变化较大,可以用只有几十万个晶体管的处理机芯片,也可用多达上亿万晶体管的复杂的处理机芯片。移动式计算机的用途广泛,随着有线和无线网络技术的迅速发展,将有更广阔的应用前景。人们可携带笔记本计算机、PDA 和无线终端等外出办公;可在移动过程中用 PDA 或无线终端上网用电子邮件和其他人通信;当移动式计算机和全球定位系统(GPS)相结合时,可在汽车中显示汽车当前的位置

和行驶路径的地图，可使野外人员知道自己所在地点的精确位置与周围情况，如果移动计算机能有语音输入与输出功能，则可和其他人通话。

参考文献

Talukder A K, Yavagal R. Mobile computing: technology, applications and service creation. McGraw-Hill Communications Engineering, 2007 （李三立）

yidong shuju guanli

移动数据管理(mobile data management)

移动计算环境中的数据管理技术，它能满足人们在任意地点任意时间访问任意数据的需要。它涉及数据库技术、分布式计算以及移动通信等多个学科领域。在移动计算模式下，计算平台的多样性、数据模型的异构性以及计算环境的复杂性给移动数据管理带来了困难与挑战。

目前移动数据管理的研究主要集中在以下几个方面：

首先是**移动数据库技术**。从数据库技术的发展来看，传统的集中式数据库及分布式数据库技术均是基于有线网络和固定主机的。这些技术都采用了一些隐含的假设，如固定网络连接、对等通信代价、主机节点固定不变等。但在移动计算环境下，上述假设条件不再成立。移动计算环境所具有的主机移动性、有限通信带宽、频繁断接性、网络通信的非对称性、电源的有限性等特点，使得传统分布式数据库中的方法和技术不能直接应用于移动数据库。移动数据库的主要研究内容包括移动事务处理技术、移动复制与缓存技术、移动数据广播技术、移动查询处理与查询优化技术等。

其次是**嵌入式/微小型数据库技术**。随着智能移动终端的普及和人们对移动数据实时处理和管理要求的不断提高，嵌入式/微小型数据库技术越来越体现出其优越性。但移动设备所具有的计算能力小、存储资源不足、带宽有限以及闪存存储等特性，使得直接裁剪传统数据库系统而得到的嵌入式/微小型数据库并不能适用嵌入计算环境的需要。嵌入式/微小型数据库技术主要研究内容包括数据压缩、RAM 的使用、存取规则、基本操作系统和硬件的支持及稳定存储等。嵌入式/微小型数据库技术目前已经从研究领域向广泛的应用领域扩展，各种微小型数据库产品纷纷涌现。

上述两类技术有时合称为**嵌入式移动数据库技术**。

第三是**移动对象管理技术**。移动对象/用户是移动计算环境下的运行主体，因而如何实施对移动对象/用户的有效管理便成为这一领域的研究热点，即移动对象数据库(moving objects databases，简称MOD)技术。移动对象数据库是指对移动对象(如车辆、飞机、移动用户等)及其位置进行管理的数据库。移动对象管理技术主要研究内容包括移动对象建模、移动对象索引、移动对象查询、移动对象聚类、移动对象不确定性等。

第四是**基于位置的信息服务**。在传统的数据处理中，数据对象的地理特性通常被人们忽略了，但在移动计算环境下，这种“位置透明”性则被“位置相关”的特性所取代。移动性是移动计算系统有别于传统分布式系统的最为重要的特征，而基于位置的信息服务则是移动信息系统最重要的功能之一。基于位置的服务(location-based service，简称 LBS)是指系统能够根据用户当前位置的变化提供与其位置相关的数据信息。

移动对象管理技术在许多领域同样展现出了广阔的应用前景。在军事上，移动对象数据库可以回答常规数据库所无法回答的查询。在民用领域，利用移动对象数据库技术可以实现智能运输系统、出租车/警员自动派遣系统、智能社会保障系统以及高智能的物流配送系统。此外，移动对象管理技术还在电子商务领域有着广泛的应用前景。

参考文献

1. 孟小峰，丁治明. 移动数据管理. 北京：清华大学出版社，2009

2. Guting R H, Schneider M. Moving objects databases. Morgan Kaufmann, 2005 （周立柱　孟小峰）

yidong tongxinwang

移动通信网(mobile communication network)　参与通信的一方或双方可以在其作用范围内随意移动并进行通信的网络，而且在参与通信的双方中至少有一方处于运动中或暂时停留在某一非预定的位置上。移动通信网按使用对象、用途、经营方式、频段、制式、入网方式等有不同分类方法。如按使用对象分，可分军用、民用；按用途和区域分，可分陆上、海上、空间；按经营方式分，可分为公众网、专用网等。

典型的移动通信网组成方案如图 1 所示。它由移动通信交换局(MTX)、基站(BS)、移动台(MS)和局间或局站间的中继线组成。移动台和基站、移动

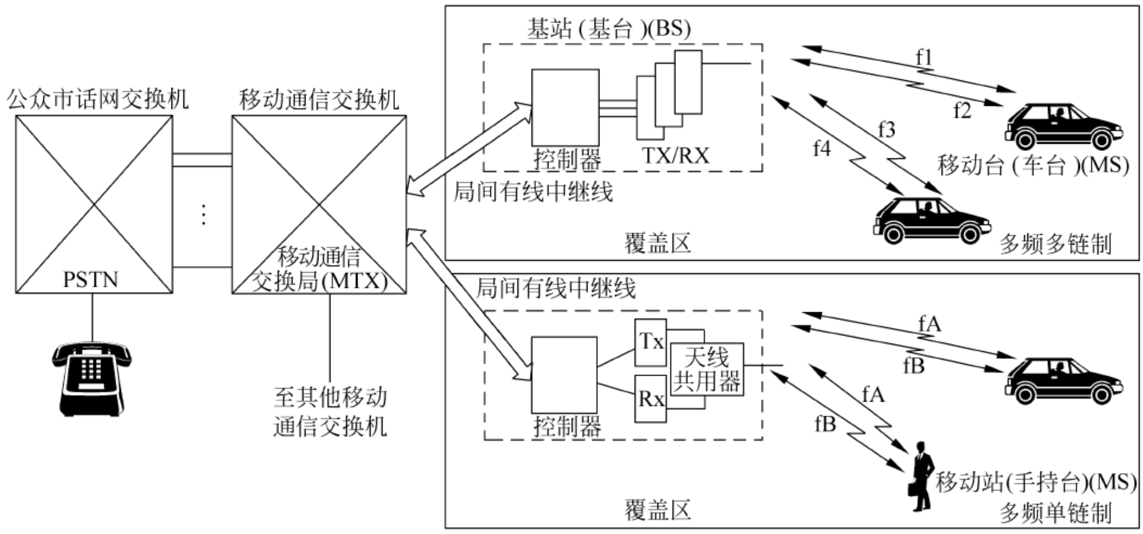


图 1　移动通信网的组成

台和移动台之间采用无线传输方式。基站与移动通信交换局,移动通信交换局与有线网(PSTN)之间一般采用有线(中继线)方式进行信息传输。移动交换局和基站担负信息交换和接续以及对无线频道的控制等。基站与移动台都设有收发信机、收发信共用装置和天线、馈线等。每个基站都有一个有发信功率与天线高度所限定的地理覆盖范围,称为覆盖区。由多个覆盖区组成全系统的服务区。

移动通信网与有线固定网相比,具有下列特点:①采用复杂的无线电波传播模式;②干扰多而复杂;③工作环境恶劣,设备要求可靠性高,体积小、重要轻和省电等;④主要通过有效利用频率技术扩大用户容量;⑤组网方式多样,网络结构和信令方式比较复杂。

移动通信的基本技术主要包括:①调制技术;②移动信道中电波传播的模式及抗衰落措施;③抗干扰措施;④组网技术;⑤移动台的小型化。

参考文献

1. 田翠云, 赵荣黎, 蒋忠涌. 移动通信系统. 北京: 人民邮电出版社, 1990

2. 郭梯云, 邬国扬, 张厥盛. 移动通信. 西安: 西安电子科技大学出版社, 1995　(史美林　英春)

yidong zhinengti

移动智能体(mobile agent)　一类具有移动能力的智能体(参见多智能体系统),即在网络环境下可以从一个节点移动到另一个节点,并在目标节点上运行。移动智能体的开发需要解决几个方面问题。首先,提供支持智能体移动及其代码运行的基础设施,如软件模型、运行引擎、安全和通信设施、事件服务等;其次,提供程序设计语言来支持移动智能体的构造和实现,包括语言设施、软件开发包、解释器和编译器等等;最后,解决由于移动计算带来的可信、信誉和安全等一系列问题。

为了支持移动智能体的开发和运行,人们已经提出了诸多移动智能体程序设计语言和支撑环境。根据程序设计语言的特点,它们大致可以分为 3 类:①解释性语言,代表性成果包括: General Magic 开发的 Telescript、Agent TCL/D'Agents 等; ②基于 Java 或者其他面向对象语言,代表性成果包括: IBM 开发的 Aglets、Mitsubishi Electric ITA 开发的 Concordia、Mole、Jade 等; ③基于 COBRA 技术,代表性成果包括: Grasshopper 开发的 IKV++ 、ObjectSpace 开发的 Voyager 等。

移动智能体技术的优点: ①减少网络流量　移动智能体可以将数据或计算移到目标计算节点上处理,减少客户/浏览器与服务器之间的频繁请求应答和中间处理数据的传送,从而有效降低网络负载。②促进并行处理　通过移动可以将智能体移到不同的计算节点和基础设施上并行执行,从而加强了计算资源的共享,推动计算的并行化。③提高系统的灵活性和适应性　移动智能体可以根据计算节点或者网络环境的变化,动态地移动到不同的计算节点上来运行,从而确保系统能够有效应对变化,使得系统具有更强的灵活性。

移动智能体技术可有效支持网络环境下的数据

查询、事务处理、环境感知等，因而可广泛应用于以下一类应用的开发，包括信息检索、网络管理、远程控制、动态系统等。

参考文献

Picco G P. Mobile agents：An introduction. Microprocessors and Microsystems，2001，25(2)：65-74

（毛新军）

yidong zhongduan

移动终端(mobile terminal) 通过无线网络，可以在移动中使用的计算机**终端设备**。广义而言包括手机、**笔记本计算机**、**平板计算机**和便携式上网设备(参见**上网本**、**个人数字助理**)，甚至包括销售点终端(POS 机)及车载计算机等。一般情况下，移动终端是指手机或具有多种应用功能的智能手机。

早在 20 世纪 40 年代美国最大的通信公司——贝尔实验室试制出了第一部所谓的移动通信电话。但是，由于体积太大，只能把它放在实验室的架子上。1973 年 4 月美国摩托罗拉公司的工程技术人员，手机的发明者马丁·库帕研制成了一个约有两块砖头大的无线电话，实现了无线通话，这便是世界上第一部手机。从 1973 年手机注册专利，到 1985 年才诞生第一台现代意义上的、真正可以移动的电话。

随着网络和技术朝着宽带化的方向发展，移动通信产业将走向真正的移动信息时代。另一方面，随着集成电路技术的飞速发展，移动终端已拥有强大的处理能力，正在从简单的通话工具转变为一个综合信息处理平台，为移动终端开拓了更宽广的发展空间。

现代的移动终端已经拥有极为强大的处理能力(中央处理器 CPU 主频已达到 GHz 量级)、内存、固态存储介质(参见**固态盘**、**操作系统**等)可与低端台式计算机比拟，是一个完整的计算机系统，可以完成复杂的处理任务。移动终端拥有丰富的通信方式，既可以通过 GSM、CDMA、3G、4G 等无线运营网通信，也可以通过**无线局域网**、**蓝牙**和红外进行通信。

现代的移动终端不仅可以通话，拍照、听音乐、玩游戏，而且可以实现包括定位、信息处理、指纹扫描、身份证扫描、条码扫描、电子标签扫描、IC 卡扫描以及酒精含量检测等丰富的功能，成为移动办公、移动执法和移动商务的重要工具。有的移动终端还将对讲机也集成到移动终端上。

移动终端已经融入经济和社会生活中，为提高人民的生活水平，提高执法效率，提高生产的管理效率，减少资源消耗和环境污染以及突发事件应急处理增添了新的手段。

（韩承德）

yidong zizuwang Ad Hoc

移动自组网 Ad Hoc(mobile Ad Hoc network，MANET) 无线自组织网又称作 Ad Hoc 网，Ad Hoc 一词源自于拉丁语，本意是“无事先准备的、临时性的”。Ad Hoc 网是由一组带有无线收发装置的可移动节点组成的一个多跳临时性无中心网络，可以在任何时刻、任何地点快速构建起一个移动通信网络，并且不需要现有信息基础网络设施的支持，网中的每一个终端可以自由地移动，所有终端从功能上来说都是平等的，没有中心。从技术上讲，Ad Hoc 网络是一种移动通信技术和计算机网络技术相结合的网络。Ad Hoc 网络中，每个移动终端兼备路由器和主机的两种功能。一方面作为主机，终端需要运行各种面向用户的应用程序，如编辑器、浏览器等；另一方面，作为路由器，终端需要运行相应的路由协议，根据路由策略和路由表完成数据的分组转发和路由工作。在部分通信网络被破坏后，这种分布式控制和无中心的网络结构能维持剩余的通信能力，确保重要的通信指挥通畅，因而具有很强的鲁棒性和抗毁性。Ad Hoc 网络使用无线通信技术进行数据传输，由于无线传输范围有限，网络中的节点相互作为其邻居的路由器，通过多个中间节点转发实现节点间的通信，即报文通过多跳才能从源端点传输到目的节点，因此它又被称为多跳网络。

Ad Hoc 网络一般有两种结构，即平面结构和分层结构。在平面结构中，所有节点的地位平等，所以又可以称为对等式结构。而分层结构中，网络被划分为若干个独立的子网，每个子网都是一个自治系统，子网之间的连接可以通过选举子网内某一个节点充当无线 AP(access point，无线访问节点)，也可以由专用 AP 构成的分发系统负责。

平面结构的网络比较简单，网络中所有节点完全对等，原则上不存在瓶颈，所以健壮性好。它的缺点就是可扩展性差，每个节点都需要知道到达其他所有节点的路由，维护这些动态变化的路由信息需要大量的控制消息，特别是当平面结构网络的规模增加到某个程度时，大量的带宽可能会被路由协议消耗掉了。在分层结构的网络中，子网内成员的功能比较简单，不需要维护复杂的路由信息，这大大减少了网络中路由控制信息的数量，因此具有很好的

可扩展性。分层结构并不局限于同构网络之间的连接,还适用于异构网络,随着应用的扩展,不同形式的无线网络也可以通过分层结构组成 Ad Hoc 网络。总之,当网络的规模较小时,可以采用简单的平面式结构;当网络的规模增大时,应该采用分层结构。

与其他传统通信网络相比,Ad Hoc 网络的主要特点是独立组网,不依赖于任何预先架设的网络设施;无中心和自组织性,节点可以随时加入和离开网络,网络的健壮性和抗毁性很好;网络拓扑动态变化,变化的方式和趋势都难以预测;有限的无线传输带宽等。移动自组织网络通常用在没有或不便利用现有网络基础设施的情形中,例如军事通信、移动会议、紧急服务和灾难恢复、连接个域网和传感器网络等。

参考文献

1. 于宏毅. 无线移动自组织网. 北京:人民邮电出版社, 2005

2. http://www. anywlan. com /Article /2008/20080404184535_3472. shtml

3. http://wenku. baidu. com/view/880ef2c2d5bbfd0a795673cc. html

(程时端)

yichuan suanfa

遗传算法(genetic algorithm) 一类模拟生物进化过程与机制的随机优化与搜索算法。它采用简单的编码技术来表示各种复杂的结构,并通过对一组编码表示进行简单的遗传操作和优胜劣汰的自然选择来指导学习和确定搜索的方向。遗传算法的操作对象是种群,种群中每一个个体都对应问题的一个解。从初始种群出发,采用基于适应值比例的选择策略在当前种群中选择个体,使用杂交和变异来产生下一代种群。如此模仿生物的进化一代代演化下去,直到满足期望的终止条件为止。

遗传算法最早由美国密歇根(Michigan)大学的 John H. Holland 教授提出,起源于 20 世纪 60 年代对自然和人工自适应系统的研究。20 世纪 70 年代,De Jong 基于遗传算法的思想在计算机上进行了大量的数值函数优化计算实验。在一系列研究工作基础上,20 世纪 80 年代由 D. E. Goldberg 进行归纳总结,形成了遗传算法的基本框架。

作为随机优化与搜索算法,遗传算法有如下特点:①遗传算法的操作对象是种群,因而具有良好的并行性。②在优化与搜索过程中,遗传算法只需利用目标函数的取值信息,而无须梯度等高阶信息,因而适用于大规模、高度非线性的多峰函数优化以及组合优化,具有很强的通用性。③遗传算法操作的种群是经过编码的,目标函数可解释为编码化个体的适应值,因而具有良好的可操作性与简单性。④遗传算法属于一种自适应随机搜索技术,其选择、杂交、变异等运算都是以一种随机的方式来进行的,从而增加了其搜索过程的灵活性。

遗传算法作为一种新的**计算智能**技术,虽然在理论和方法上还有待于进一步发展和完善,但却为求解许多复杂问题提供了希望。遗传算法被人们广泛地应用于函数优化、组合优化、**机器学习**、自适应控制和**人工生命**等领域,是计算智能中的一项关键技术。近年来遗传算法的应用研究显得格外活跃。随着应用领域的扩展,遗传算法的研究出现了几个引人注目的新动向:一是基于遗传算法的机器学习,这一新的研究课题把遗传算法从离散空间的优化搜索算法扩展到具有独特的规则生成功能的机器学习算法。二是遗传算法正日益和**人工神经网络**、模糊推理以及混沌理论等其他计算智能方法相互渗透和结合,这对开拓新的计算智能技术将具有重要的意义。三是并行处理的遗传算法的研究十分活跃。这一研究不仅对遗传算法本身的发展,而且对于新一代智能计算机体系结构的研究都是十分重要的。四是遗传算法和人工生命的研究领域正不断渗透。五是遗传算法和进化规划以及进化策略等进化计算理论日益结合。

参考文献

1. Holland J H. Adaptation in natural and artificial systems. Ann Arbor: University of Michigan Press, 1975

2. Goldberg D E. Geneticalgorithms in search, optimization and machine learning. Reading, MA: Addison-Wesley, 1989

3. 周明,孙树栋. 遗传算法原理与应用. 北京:国防工业出版社,1999

(陆玉昌)

yitaiwang

以太网(Ethernet) 采用带碰撞检测的载波侦听多址访问(CDMA/CD)方法进行介质访问控制的一种局域网。1972 年起,Xerox 公司开发了一个采用 CSMA/CD 方法、速率为 2.94 Mb/s 、运用一根同

轴电缆把众多的 Xerox 公司发明的个人工作站、打印机和服务器连接起来的网络,取名为“Ethernet”。接着,DEC 公司和 Intel 公司加入了以太网的规范的制定,1980 年 Xerox 公司、Intel 公司和 DEC 公司一起制定和发表了 1.0 版本的以太网规范(10 Mb/s),随后加入了一些技术变更、错误改正和小的改进,1982 年发布为第 2 版。这些规范被业界称为 DIX 以太网规范。DIX 规范被 IEEE 标准委员会采纳并作了一些小修改后,在 1985 年成为 IEEE 802.3 标准(参见**局域网协议标准**),并不断向前发展。

最初的以太网采用同轴电缆作为传输介质,各个站点通过分接头连接到同轴电缆上,共享该传输介质,构成总线式拓扑结构,因此,常称这个介质为总线。信号的速率为 10 Mb/s,采用曼彻斯特编码(参见**数据编码**)。以太网的数据传送以**数据帧**形式进行。这个帧业界称为以太帧。它由长度固定的目的地址、源地址、数据长度、帧检验序列字段以及长度变化的数据和填充等字段组成,总长度不是固定的,最短为 64 字节,最长为 1518 字节。以太帧格式如图 1 所示,其中前导和起始定界符是物理层的信息,不是以太帧的字段。站点发送数据时,以以太帧的形式把数据发送到介质上。该帧沿着介质传播到各个站点,那些地址与以太帧的目的地址字段相符的站点把该帧接收下来,其余站点丢弃该帧,这样就完成了一次数据传输。一个站点发送以太帧是按照带有冲突检测的载波侦听多址访问(CSMA/CD)介质访问控制方法进行的(参见**局域网介质访问控制方法**)。由于电磁波的传播需要时间,在最坏情况下,一个站点的发送要经过总线端到端的传播延迟的两倍时间(称为路径时延(path delay value),又称为冲突窗口)才能发现是否有冲突。这个窗口定义为 512 位(64 字节),对于以太网这个窗口对应的时间为 51.2 μs(0.1 μs × 512 = 51.2 μs)。这个窗口限定了以太帧的最短长度为 64 字节,也限定了总线的最大长度,当总线长度超过了最大长度,以太网的运行变得不可靠。

字节数	7	1	6	6	2	46~1500	4
	前导	SFD	DA	SA	长度*	数据及填充	FCS

SFD:起始定界符;DA:目的地址;
SA:源地址;FCS:帧检验序列;*值大于 1536,这个字段的意义是类型

图 1 以太网的帧格式

随着以太网的发展,它使用的传输介质类型不断扩展。最初使用粗同轴电缆,随后,细同轴电缆、非屏蔽双绞线、屏蔽双绞线和光纤等多种传输介质都进入了以太网。

10Base-T 使用双绞线电缆,最大网段长度为 100 m,物理拓扑结构为星型;10Base-T 组网主要硬件设备有 3 类或 5 类非屏蔽双绞线、带有 RJ-45 插口的以太网卡、集线器、交换机、RJ-45 插头等。

当越来越的站点加入到以太网时,网络上的传输量激增,冲突加剧,降低了网络的有效带宽。为解决这个问题,发展出了**交换式以太网**。引入了**交换机**,把一个冲突域分隔成多个冲突域。减少每一网段上的站点数,降低了访问冲突,增加每个站点享用的平均带宽,提高了有效带宽。

随着网络的发展,传统标准的以太网技术已难以满足日益增长的网络数据流量速度需求。1995 年 3 月 IEEE 宣布了 IEEE 802.3u 100BASE-T 快速以太网标准(Fast Ethernet)。快速以太网支持 3、4、5 类双绞线以及光纤的连接,能有效地利用现有的设施。但由于快速以太网仍是基于 CSMA/CD 技术,当网络负载较重时,会造成效率的降低,交换技术可以提升以太网的吞吐率。

千兆以太网技术继承了传统以太技术价格便宜的优点,采用了与 10 M 以太网相同的帧格式、帧结构、网络协议、全/半双工工作方式、流控模式以及布线系统,可与 10 M 或 100 M 的以太网兼容。IEEE 标准支持最大距离为 550 m 的多模光纤、最大距离为 70 km 的单模光纤和最大距离为 100 m 的铜轴电缆。千兆以太网填补了 802.3 以太网/快速以太网标准的不足。

万兆以太网规范包含在 IEEE 802.3 标准的补充标准 IEEE 802.3ae 中,它扩展了 IEEE 802.3 协议和 MAC 规范,使其支持 10 Gb/s 的传输速率。除此之外,通过广域网接口子层(WAN interface sublayer, WIS),万兆以太网也能被调整为较低的传输速率,如 9.584640 Gb/s (STM-64 或 OC-192),这就允许 10 千兆位以太网设备与同步光纤网络(SDH 或 SONET)传输格式相兼容。

最新的以太网标准已经是 40 G/100 G 高速以太网标准,400 G/1T 的超高速以太网也在制定中。

参考文献

1. Charles E. SpurgeonEthernet: The definitive guide. O'Reilly & Associates, 2000

2. IEEE Std 802.3-2008 section 1. IEEE Com-

puter Society, 2008 （徐伟民　王能　金耀辉）

yibu chuansong moshi

异步传送模式（asynchronous transfer mode, ATM） ATM以分组交换为基础并结合**电路交换**的高速特性而形成的一种技术。ATM技术是20世纪80年代后期由国际标准组织ITU-T针对电信网支持宽带多媒体业务而提出的。它将数据、图像和话音（言语）等信息分解成定长的信息分组，并在信息分组的前面装上含有地址等控制信息的信头称为信元，同时以信元为单位进行标记复用。由于只要获得信元头即可插入信息并传送出去，而且信息的插入位置是非周期性的，因而称之为异步传送模式。

ATM的信元结构

信元实际上就是小分组，ATM的信元具有固定的长度，为53B，其中5B是信头，48B是信息字段。信头的结构如图1所示。用户网络接口（UNI）和网间接口（NNI）上信头结构稍有不同。GFC是一般流量控制字段，用于控制同一接口上多个终端所发送的业务量，以减少可能出现的网络过载。

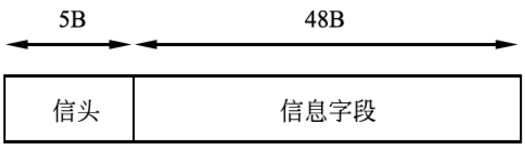


图1　信头的结构

ATM的分层结构

ATM网的功能十分单纯，它仅有**开放系统互连OSI参考模型**的第1层物理层和ATM层，与其他高层无关。信元在ATM层上传递，其标记在呼叫建立时分配，呼叫结束时释放，它是面向连接层的业务。

在ATM中，将开放系统互连OSI参考模型的第1层细分为物理层、ATM层和ATM自适应层（AAL）。物理层承运信元流的负载。ATM层提供传送信元所需的最低功能。AAL层的主要作用是将第2层以上的用户信息分段装配成信元，并进行流控和差错控制。网络只提供到ATM层为止的功能。AAL层的功能由用户本身提供或由网络与外部的接口提供。

我国在1997年建成并开放帧中继/ATM数据网（CHINAFR/ATM）。但由于ATM采用的是信令协议，所以与其他网络的互通互联能力差。近年来ATM已经逐步被互联网取代。

参考文献

程时瑞. 综合业务数字网. 北京：人民邮电出版社，1993

（马妍）

yibu shuju lianlu kongzhi xieyi

异步数据链路控制协议（asynchronous data link control protocol） 主要指面向字符传输的数据链路协议，用于早期较为简单的数据通信环境。

XON/XOFF是一种典型的异步字符通信流量控制协议，其中XON采用ASCII字符集中的控制字符DC1，XOFF采用ASCII字符集中的控制字符DC3。当通信线路上的接收方发生过载时，便向发送方发送一个XOFF字符，发送方接收XOFF字符后便暂停发送数据；等接收方处理完缓冲器中的数据后，再向发送方发送一个XON字符，以通知发送方恢复数据发送。在一次数据传输过程中，XOFF、XON的周期可重复多次，但这些操作对用户来说是透明的。XON/XOFF方法广泛使用于KERMIT和XMODEM等文件传输协议中。

异步通信协议（asynchronous communication protocol）或称异步传输（asynchronous transmission）也可以归为**异步协议**。该协议中字符与字符之间的间隔时间是不固定的（即字符之间是异步的），在每个字符的起始处开始对字符内的比特实现同步，由于发送器和接收器中近似于同一频率的两个约定时钟能够在一段较短的时间内保持同步，所以可以用字符起始处同步的时钟来采样该字符中的各比特，而不需要每个比特再用其他方法同步。"起-止"式通信规程便是异步协议的典型实例，它是靠起始位（逻辑0）和停止位（逻辑1）来实现字符的定界及字符内比特的同步的。异步协议中由于每个传输字符都要添加诸如起始位、校验位、停止位等冗余位，故信道利用率很低，一般用于数据速率较低的场合。

起-止式异步通信规程的特点：

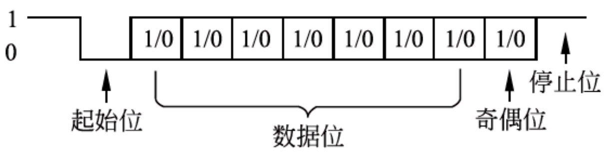
（1）每一个字符独立地发送，字符间的间隔是任意的。每个字符的组成部分：

起始位　1位，一个字符的开始；

数据位　5～8位（最低位在前）；

奇偶检验位　1位（可选）；

停止位　1、1.5或2位，一个字符的结束。



(2) 每个字符以起始位和停止位加以分割,故称起-止式。

(3) 字符中各个比特用固定的时钟频率传输,但字符间采用异步定时,字符间的同步利用起始位实现,收、发时钟只要在一个字符的时间内保持同步(误差 <7%)即可,不要求两个时钟频率精确一致。

参考文献

1. Forouzan B A. 数据通信与网络. 2 版. 吴时霖,周正康,吴永辉,等译. 北京:机械工业出版社,2002

2. Gilbert Held. 数据通信. 6 版. 戴志涛,卞佳丽,郑岩,译. 北京:人民邮电出版社,2000

(许勇 张凌)

yichang chuli

异常处理(exception handling) 在**程序设计语言**中用于描述异常与异常处置而用的语言机制。语言中提供的异常处理机制一般包括引发机制和处理机制两部分。

有几种解决异常处理的方案,一种是把异常处理看作针对非经常发生的事件(不一定是错误)的一种正常程序设计技巧,异常情况发生时,便由异常处理程序处理,处理结束时,控制还可以回到发生异常的位置。因此,也可以利用异常处理来实现一些修补工作。当修补结束时,又继续程序的正常执行。另一种则只限于异常情况发生了一些称作"错误"(或者是结束条件)的事件。这便表示,在某一程序单位中发生异常情况时,执行就告终止,把控制转到异常处理程序,但以后就不再回到发生异常情况的位置,处理程序可以决定重新启动有关的程序段,但不是简单的恢复。无法恢复时,保留现场信息供分析用。

参考文献

徐家福. 系统程序设计语言. 北京:科学出版社,1983

(程虎)

yinzi fenjie

因子分解(factoring) 两正整数 a,b,若 b 能除尽 a,则称 b 为 a 的因子,记为 $b|a$。一个大于等于 2 的自然数若仅以 1 和自身为其因子,称其为素数。任一大于等于 2 的自然数都能唯一地表成素数幂的乘积(算术基本定理)。设 $n\geqslant 2$ 为自然数,则 n 可唯一地因子分解为 $n=p_1^{a_1}p_2^{a_2}\cdots p_s^{a_s}$,其中 $p_1<p_2<\cdots<p_s$ 为素数,$a_i(1\leqslant i\leqslant s)$ 为正整数,称此式为 n 的标准因子分解式。由于计算机技术的发展和数论在密码学等领域的应用,使研究大整数因子分解算法成为很活跃的研究课题(参见**计算数论**)。 (裴定一)

yinying shengcheng

阴影生成(shadow generation) 一种生成阴影的技术。阴影是由于光源被遮挡而形成的较暗区域。作为一种常见而又重要的视觉效果,阴影不但能够增强三维场景的真实感,而且能提供一些辅助信息帮助观察者判断景物的空间位置关系。阴影生成的本质问题是求解绘制方程中的可见性。根据阴影效果的不同,阴影可以分为两大类:硬影(hard shadow)和软影(soft shadow)。硬影边界清晰,是由点光源被不透明物体遮挡所形成;软影也称柔和阴影,有一个从明到暗的过渡区域,界限模糊,是由面光源产生,包括本影区(umbra)和半影区(penumbra)两部分:完全不被光源直接照到的区域称为本影区,部分地被直接照到的区域(即明暗过渡区域)称为半影区。

整体光照明算法能够自然地绘制出各种阴影效果。基于辐射度的绘制算法在计算形状因子过程中考虑了光源的遮挡,能够生成阴影效果。基于光线跟踪的绘制算法通过向光源发射阴影光线探测是否位于阴影区域中。如果是面光源,则需要发射多条阴影测试光线,累积结果。在游戏和虚拟现实等对绘制效率要求较高的三维图形应用中,通常使用**图形处理器**(GPU)加速阴影绘制,其中代表性算法有阴影图(shadow map)算法和阴影体(shadow volume)算法。

阴影图算法是一种两步处理的图像空间算法。首先从光源对场景进行第一遍绘制,得到从光源可见像素的深度纹理图,称为阴影图(shadow map);然后从视点对场景进行第二遍绘制,绘制过程中将当前片元转换到光源空间,通过查询第一遍得到的深度纹理得到该片元与光源的距离,进行距离的比较,从而确定该点是否处于阴影中。这种方法的优点是效率与场景复杂度关系不大,便于硬件实现,但是由于深度纹理图只是对场景的一种离散表示,会在阴影边缘出现锯齿现象。近年来提出多种办法解决这一问题。

阴影体算法是一种处理点光源产生硬影的对象空间算法,其基本思想是以点光源为中心,将遮挡光源的景物逐点向后延伸形成阴影体,阴影体内的点

都处于阴影之中。阴影体的构造步骤如下：首先以光源为视点，提取景物的轮廓线；然后将每一段轮廓线向远处延伸，形成空间四边形；最后将这些空间四边形构成的柱状体的两头封闭起来，得到阴影体，通过判断一个具体的像素是否位于阴影体中以确定其可见性。运用阴影体绘制步骤如下：首先不考虑光照，将场景渲染到帧缓存中；然后将阴影体渲染到模板缓存（stencil buffer）中，根据渲染的表面朝向和深度测试增减模板缓存的值，构造阴影掩膜。最后再一次渲染场景，根据阴影掩膜代表的阴影信息，计算光照。实际实现过程中可以通过模板缓存对其进行累积计算。这种方法能够准确地计算场景的阴影，但是由于每一帧都需要实时计算物体的轮廓变化，这使得当场景中的物体几何复杂度比较高的时候，算法效率较低。

为了绘制面光源产生的软影效果，人们对阴影体和阴影图算法进行扩展，提出了软阴影体和软阴影图等近似算法。

参考文献

1. Franklin C. Crow: Shadow algorithms for computer graphics. ACM SIGGRAPH Computer Graphics, 1977, 11(2): 242-248

2. Williams L. Casting curved shadows on curved surfaces. ACM SIGGRAPH Computer Graphics, 1978, 12(3): 270-274

3. Guennebaud G, Barthe L, Paulinm M. Real-time soft shadow mapping by backprojection. In: Eurographics Symposium on Rendering, 2006

4. Assarsson U, Akenine-moller T. A geometry based soft shadow volume algorithm using graphics hardware. ACM Trans on Graph, 2003, 22(3): 511-520

（刘新国　吕伟伟）

yinpin bianma

音频编码（audio coding）　由麦克风采集的音频信号是一个连续的模拟信号，通常采用脉冲编码调制 PCM 方式进行基础的数字化编码；针对不同的用途，还可以进行二次编码，例如适合窄带与宽带话音通信的 ITU-T G. 711 和 G. 722 编码标准等；适合音乐交换的 AC-3 标准和 MP3 标准等。

在音频的数字化 PCM 编码过程中，采样频率和采样精度两个指标最为关键。由于人耳能够感觉到的最高频率为 20 kHz 左右，根据奈奎斯特采样定理，采样频率必须大于 40 kHz，常见的激光唱片 CD 的音频采样频率为 44. 1 kHz；采样精度决定了数字音频信号的信噪比，CD 音频采用 16 bits 量化，一个双声道 CD 质量的 PCM 编码音频数据速率达到 44. 1 kHz × 16 bits × 2 = 1411. 2 kbps。为了存储 1 个小时以上 CD 唱片质量的音频文件，需要的 CD 光盘或硬盘容量超过 605 MB。

为了满足各种传输和存储设备的要求，各种低质量的压缩编码算法出现，例如 WMA，OGG，MP3 等。而用于通信的 G. 711 窄带语音编码标准的采样频率为 8 kHz，G. 722 宽带语音编码标准的采样频率为 16 kHz，极大地降低了传输速率。

参考文献

胡道元. 计算机网络（高级）. 北京：清华大学出版社，1999

（周杰　张凌）

yinpin shipeiqi

音频适配器（audio adapter）　一种实现声波和数字信号的相互转换的硬件转换器。又称声卡。作为进行声音处理的适配器，它是**多媒体技术**中最基本的组成部分。它能够将来自话筒、磁带等输入设备的原始模拟声音信号经过**模数转换器**处理后存储到数字存储设备中，或将数字信号经过**数模转换器**处理后传输到耳机、扬声器、扩音机、录音机等输出设备，或将存储的数字信号直接通过乐器数字接口（musical instrument digital interface，MIDI）处理器使乐器发出声音。

音频适配器分类　主要分为板卡式、集成式和外置式三种类型。

（1）板卡式　早期的板卡式音频适配器主要采用 ISA 接口，由于这种接口总线的带宽较低、功能单一，且同时占用过多系统资源，已逐渐被外围部件互连（PCI）接口的音频适配器所取代。

（2）集成式　此类产品集成在主板上，它不占用 PCI 接口且成本低，在普通计算机中被广泛使用。由于这种音频适配器基本不占用额外的空间，且能够满足大部分用户对音频适配器的性能要求，因而目前在应用中占有主导地位。

（3）外置式　它通过通用串行总线（USB）接口与计算机连接，使用方便，是一种便携式的音频适配器。这类产品通常应用于特殊环境。

音频适配器硬件　包括以下几个部分：

（1）声音控制芯片　可以从**输入设备**中采集声音模拟信号，通过模数转换器，将模拟信号通过采样和量化转换成数字信号并存储到计算机中。重放

时,数字信号通过数模转换器还原为模拟波形,经过功率放大后送到扬声器放音。

(2) 数字信号处理器　用来对数字信号进行采集、分解、变换、压缩、综合等各种处理。它可以处理有关声音的命令、执行压缩和解压程序、增加特殊声效等。**数字信号处理器**可以有效地减轻中央处理器(CPU)的运行负担,加速多媒体软件的执行速度。

(3) 语音合成芯片　一般采用频率调制(frequency modulation,FM)合成芯片或者波表合成芯片合成声音。波表合成能够比 FM 合成获得更加逼真的效果。

大多数音频适配器的主要连接端口有:

(1) 线型输入接口　用来输入外部声音源的声音信号,通过计算机控制对输入信号进行录音或混音然后存储到计算机硬盘上。该端口通常用于外接辅助声音源,从话筒、磁带式录音机等系统输入声音。

(2) 线型输出接口　将声音信号从音频适配器传输给计算机外部设备。可以输出到音响、耳机等声音播放设备。

(3) 话筒输入端口　用于连接话筒录制声音。

(4) 扬声器输出端口　属非必需的,如果该端口和线型输出接口同时存在,则线型输出接口的信号经过放大后传输到扬声器输出端口。

(5) 乐器数字接口(MIDI)　连接计算机外部的 MIDI 设备,实现 MIDI 信号的直接传输。

(6) 游戏杆接口　连接任何标准的游戏杆或游戏控制柄配合游戏软件。

(7) 内部连接接口　通过带状电缆将内部光碟驱动器与音频适配器直接连接。这种连接方式可以将声音信号直接从光碟驱动器(参见**光存储器**)传输给音频适配器,再通过计算机音箱进行播放。

音频适配器的质量衡量标准　通常用两个指标来衡量:频率响应范围和总谐波失真。音频适配器的频率响应是指音频系统能够听得见的振幅,以及进行记录和播放的频率范围。多数音频适配器的频率响应是 30Hz 到 20KHz,通常范围越宽,性能越好。总谐波失真衡量音频适配器的线性度,也就是频率响应曲线的直线性。任何非线性因素都将导致谐波形式的失真,失真度越小越好。　(陶建华)

yinpin xinhao shibie

音频信号识别(audio signal recognition)　用计算机对人类可听频率范围内的音频信号(包括语音、音乐等)进行分析和分类的技术。通过音频信号的识别,可以得到人们日常发声的各种属性,包括语音内容、说话人的身份、所用语言的种类和各种音频事件等;可以得到音乐信号的属性,如旋律、风格、歌手身份、乐器种类等信息;可以得到自然界中存在的其他声音的属性,从而提供对音频信息的操控和管理手段。音频信号识别是一个综合性的、多学科交叉的研究方向,涉及声学、听觉、信号处理、音乐学、**模式识别**、**人工智能**、概率论、信息论、**自然语言处理**和认知科学等。主要的任务包括语音识别,说话人识别和语种识别,音乐信号识别和音频事件检测等。

语音识别

自动语音识别可以把语音内容自动转化成文字。语音识别的难易程度受到多方面因素影响,包括词汇量的大小,发音方式(如孤立词的、连续的),是否跟说话人相关,环境噪声的情况等。一个典型的语音识别系统包括前端信号处理、统计声学模型,语言模型和解码器等模块。其中,前端信号处理主要是从输入语音信号中提取用于声学建模的各种特征,并进行各种增强鲁棒性的处理,如降噪等;声学模型主要是采用的基于隐马尔科夫方法,近年来出现了基于神经网络模型的方法;语言模型主要采用的是简单高效的基于统计的 n 元文法;解码器则是根据声学和语言模型以及提供两者基本单元间的映射的发音词典,来寻找对于给定语音信号的最可能的词串。语音识别技术有着广阔的应用前景,如口语文档的丰富转抄和检索,基于语音的人机交互等。

说话人识别

参见**说话人识别**。

语种识别

语种识别可以分为语种辨识和语种确认。语种辨识是指辨识待测输入语音属于某语种集合中的哪一种语言;而语种确认则是判断待测输入语音是否属于某一特定语种。语种辨识中使用了发音参数、声学特征、韵律、词汇知识等多种语音和语言知识源。目前的主流解决方法有音子识别后接语言模型方法和向量空间描述方法。语种确认通常看成是一个假设检验问题,并可以借鉴语种辨识中的建模和分类器设计方法。语种识别技术可以应用到多语种语音识别、对话、翻译和文档检索系统中,也可以在情报和安全系统中发挥作用。

音乐信号识别

音乐信号识别主要是对音乐的低层信号进行自

动分析,识别出音乐的各种高层属性,如音高、节奏、旋律、音乐风格和音色等。音乐信号识别系统或多或少利用了人耳的听觉模型,特别是人耳的音高感觉模型。音乐信号是时变的,大多数音乐信号识别系统先采用联合时频分析,然后进行特征提取,最后利用模式识别方法。音乐信号识别技术可以应用到音乐检索、乐器制造和计算机辅助音乐教育等领域。

音频事件检测

音频事件检测,或音频结构化,是对各种音频流中的特定事件进行分段和分类的研究。采用基于音频内容的方法来对多媒体进行结构化,研究得较多的主要是广播新闻和体育比赛视频数据。音频分段方法主要有基于能量的、基于模型的和基于距离度量的等方法。音频分类基本上采用的是模式分类和统计机器学习的方法。音频分类器的建立思路是,对音频进行分析,提取各种特征,基于一定数量的标注数据建立统计分类模型。

参考文献

1. 颜永红. 语言声学进展及其应用. 应用声学, 2009, (2), 81-89

2. Casey M, Veltkamp R, Goto M, Leman M, Rhodes C, Slaney M. Content-based music information retrieval: Current directions and future challenges. Proceedings of the IEEE. 2008, 96(4), 668-696

3. Benesty J, Sondhi M M, Huang Y (eds.) Springer handbook of speech processing. Springer, 2008

(颜永红)

yinshi qumian

隐式曲面(implicit surface) 由满足函数约束关系$\{(x,y,z)|f(x,y,z)=0\}$的点构成的集合。与显式定义的**参数曲面**(如B样条曲面)不同,隐式曲面上的点一般不能直接计算。

常用的隐式曲面有 Metaball 曲面、卷积曲面、变分曲面、分片代数曲面等。与目前在**几何造型**中占主导地位的样条参数曲面相比,隐式曲面既可表达高阶连续的光滑外形,又能描述具有任意拓扑的几何形状。此外,隐式曲面在曲面求交、曲面光滑拼接、变形物体表示等方面也具有优势。隐式曲面应用中的主要不便之处是其外形难以交互控制、曲面显示算法复杂。目前,隐式曲面的显示方法主要包括光线跟踪、多边形化和基于粒子系统的算法等。多边形化算法和基于粒子系统的算法用点、三角片或小圆盘逼近隐式曲面。这类显示算法效率高,但是容易忽略曲面的细节;光线跟踪方法可以以像素精度高质量地显示隐式曲面,但是计算量较大。随着**图形处理器**的快速发展,已经能够实现常用隐式曲面的交互式绘制或实时绘制。这为隐式曲面外形的交互编辑和控制提供了显示基础。

为了充分发挥隐式曲面,特别是分片代数曲面在几何造型中的优势,人们研究了参数曲面和分片代数曲面之间相互转化的理论和方法。有理多项式参数曲面均可以通过消去参数,实现隐式化;反之,二次代数曲面和具有一个奇异点的代数曲面可以有理参数化,其他代数曲面通常不能有理参数化。由于代数曲面没有边界,因此,在代数曲面造型中,通常采用分片代数曲面进行拼接的方法,实现复杂物体的造型,如分片二次曲面、定义在单形上 Bernstein 多项式分片代数曲面、代数B样条曲面等。分片代数曲面造型的理论基础和方法仍处于发展的过程中。随着理论和算法上的突破,代数曲面可望作为一种新形式的曲面描述方法,在CAD领域中发挥重要作用。

隐式曲面在柔性和可变形物体、边界不清晰的模糊物体(液体、云、烟雾、火焰等)等模拟方面具有优势。几十个至数以百计的 Metaball 曲面在一起可以高质量地表示人体模型,并且这样的人体模型在运动时,可以很好地表现肌肉的变形。此外,利用 Metaball 曲面的融合或分离特性,可以真实地模拟流体、黏稠液体、运动的云等。所以,Metaball 曲面在计算机动画领域中获得了广泛和成功的应用,例如 Softimage XSI、3DS MAX 等著名动画软件均集成了 Metaball 曲面造型方法。

参考文献

Bloomenthal J, Bajaj C, Blinn J, et al (eds). Introduction to implicit surfaces. San Francisco, CA: Morgan Kaufmann Publishers, Inc., 1997

(冯结青)

yinshuati hanzi shibie

印刷体汉字识别(printed Chinese character recognition) 通过扫描仪、摄像设备等将印刷文本图像输入计算机,并将其中的**汉字**图像转换成其计算机内部代码的过程和技术。

印刷体汉字识别要解决各种字体、各种字号、各种汉字形态等实际应用中出现的上万个不同印刷体汉字的识别问题。需要解决的问题有:

(1) 巨大的汉字集合,简体汉字 6767 个,简繁混合汉字共 27 484 个,包括古汉字在内有 7 万~

8 万个汉字。

(2) 不同字体的多样变化,基本字体有宋、仿宋、楷、黑、圆、隶书、魏碑等,以及各种变化和美术字体,还有各种字体所有的不同变形,总数达数百种以上。

(3) 不同大小字号的汉字,1 ~ 7 号,经常出现在同一页面。

(4) 汉字文本中除包括必要的数字和标点符号(数十个)外,现代文档均为汉、英混排文本。因此,**汉字识别**必须同时解决汉、英双语识别和汉、英混排文档识别的特殊文字切分、语种判别问题。

(5) 实际印刷文本及其扫描图像中经常出现的各种噪声干扰,汉字的模糊粘连或笔画断裂不全,汉字字形的拉伸切变等几何形变等低质量文本图像的汉字识别。

(6) 摄像设备的便宜和普及,使摄像输入文档的文字识别需求大量增加。不仅包括文本的摄影图像识别,还包括随意从自然景物中摄取的文字的检测和识别。由于摄像过程引入的字符图像透视变换,极大增加了汉字字形在大小、位置、方位、字形等方面的非线性畸变,给字符的检测、切分和识别带来极大困难。

(7) 印刷文稿包括各种公式和特殊符号,尤其是在特殊的科技文献中,如微积分等特殊的数学运算符号和公式,特殊的化学分子结构式等的识别,由于其变化的不确定性和多样性,公式的分解切分会遇到很大困难,正确识别则更不易了。

除了解决单字识别问题以外,所有印刷文本的识别还必须解决好字符串的单个字符切分问题。由于字符粘连和断裂,以及多种字符(数字、标点、英文)的混排问题,使得字符切分成为制约文本识别性能的不亚于字符识别的又一至关重要的环节。

文字识别技术的研究和发展,最重要的表现为识别的鉴别能力不断提升及对上述变化的适应能力大为加强,即识别的鲁棒性不断提高。为此,不仅进行特征的选择和优化,而且在分类器上也利用二阶非线性分类器,如 MQDF 等。实验证明,这些统计识别方法较好地解决了印刷汉字识别的问题(参见**汉字识别**)。

印刷体汉字识别研究经历了从单字体识别开始,经多字体识别,识别字体不断增加,识别汉字的字符集合不断扩大,汉字识别率,尤其是对印刷质量低下的汉字的识别率得以不断提高的过程。近来,输入设备从扫描仪向摄像机发展,识别图像从二值图像的识别发展到灰度、彩色图像的识别。目前开发出的识别系统已经不仅能够同时识别几乎所有字体,而且对实际低质量的印刷文本和摄像印刷文字也具有很高的识别率。识别率可以达到 97% ~ 99% 以上,基本满足实际文本自动输入的需要。

我国有五千年悠久的文明历史,大量古籍既需要也有可能进行现代信息处理。因此,将有 5 万余不同汉字的古籍文本输入计算机进行识别的任务也摆到了我国科学工作者的面前。

参考文献

1. 张炘中. 汉字识别技术. 北京:清华大学出版社,1992

2. 陈明,丁晓青,吴佑寿. 多层次可信度指导下的自底向上的版面分析算法. 模式识别与人工智能,2003,16(1):1-6

3. 丁晓青,王言伟,等. 文字识别:原理、方法和实践. 北京:清华大学出版社,2016

(丁晓青)

yinzhiban ceshi

印制板测试(printed circuit board testing) 以多种专用设备对裸印制板和已装元器件印制板的印制线通断的正确性,绝缘性能以及逻辑功能等进行的全面测试。万用表、示波器是传统的印制板的测试工具和仪器。随着电子工业的发展,印制板的复杂程度越来越高。基于各种测试方法的印制板测试仪已日益增多,从功能来分类主要有裸印制板测试仪、已装印制板测试仪和电路板维修测试仪三大类。其中已装印制板测试仪又可分为在线测试仪(包括故障分析仪)、功能测试仪和综合测试仪。

裸印制板测试 裸印制板测试主要是判断印制板印制线通断的正确性及其绝缘性能。其测试方法通常采用一个可进行快速扫描的电桥,当被测两点之间的电阻小于量程标值时电压比较器输出 1 电平,大于量程标值时输出 0 电平,这样就可判断被测两点之间的通断情况。一般测试用电压为 0.1 ~ 10 V,绝缘测试时需 250 ~ 500 V(甚至需 1500 V)。裸印制板测试仪一般由**微型计算机**、测试电路和测试针床组成。测试针床是这类测试仪的关键部件,弹簧触针又是测试针床的关键零件。触针上加有 0.98 N(0.1 kgf)的测量力,触针应有 50 万次以上的寿命。触头的形状可根据被测板的类型选择三棱柱形、皇冠形等形状。早期的这种测试仪一般采用专用针床,后来多采用通用针床。通用针床是按标准网格在每一个网格点上放置一个弹簧触针,在更换

被测印制电路板品种时采用这种网格式触针阵列可减少针床的制作费用。裸印制板测试仪向着网格式通用针床、高测试点容量、高测试速度、高压绝缘测试的方向发展。

印制板在线测试　参见**印制板在线测试**。

印制板功能测试　将印制板看作一个完整的模块,测试其功能。测试时给被测电路板加供电电源,使其处于正常的工作状态,从电路板的输入端输入激励信号,检测输出响应。为了故障定位,要输入适当的测试码,这样才能将内部故障传递到输出端。

功能测试的编程比较复杂,必须详细了解被测电路板上元器件的电气特性,采用大内存容量的高速计算机。通常这类测试放在电路板装配流水线的最后,用来检测产品的整体功能,故障覆盖率可达95%左右。

后来出现的一种印制板综合测试仪同时具有静态在线测试和动态功能测试两种测试功能。

印制板维修测试　指不是批量生产时的印制板测试,测试的目的是找出有故障的器件,修复印制板。印制板维修测试仪一般不采用针床结构,而采用“夹子”或探笔的方式,结构紧凑,便于携带。

数字电路印制板的维修测试可以使用通用的测试仪(如逻辑分析仪、仿真器、微型计算机故障诊断仪等)。专用的印制板维修测试仪有两大类,一类是具备在线测试能力的维修测试仪,它通常有40至64个数字测试通道,每个通道至少有1 kb×4的通道存储器。可测试小规模、中规模、大规模集成电路以及半导体存储器等器件。这类测试仪有的还带有模拟测试通道,可测量电压、周期、延时等参数。另一类是采用U-I曲线法的维修测试仪。这种测试仪使用时给被测元件加上交流电压信号,用被测元件两端产生的电压降控制示波器的水平偏转,用通过被测元件的电流控制示波器的垂直偏转,这时示波器荧光屏上出现的就是元件的电压-电流关系曲线,通常称为U-I曲线。由于每种元件都有各自的U-I曲线,因此U-I曲线也称为特征波形。通过对特征波形的分析、比较就可方便地判断元器件的好坏,找到电路板故障所在。该类仪器的特点是测试时被测电路板是不加电的,各种类型的器件都可用特征波形法判断其好坏。

基于新的测试方法的仪器,比如利用数据压缩技术的特征分析仪,采用红外线技术的电路板缺陷测试仪等,都可对印制板进行维修测试。(桑光道)

yinzhiban sheji

印制板设计(printed circuit board design)
根据计算机部件的逻辑框图或电路连接图用印制电路技术实现有关元器件之间的互连,并达到预期功能和性能的设计方法和过程。

在计算机工程设计中,插件板与底板是印制板(PCB)设计的主要任务。插件板一般是具有独立功能的部件(如处理器、存储器、输入输出接口、通信部件等),或者是具有一定逻辑关系的组合部件。底板则是实现插件间电气连接与电源馈给以形成系统或大部件的重要结构。在较复杂的计算机系统中,往往还要进行一些辅助板或附加板(如匹配板、转接板、扩展板等)的设计。

印制板设计的主要内容

(1)结构设计　结构设计既要解决PCB与其他结构件相互关连的外部相关结构问题,又要确定PCB的内部结构问题。

外部相关结构设计的主要内容有:①外形尺寸(长、宽、厚度)及精度;②在整机中安装的结构形式,有关的安装尺寸及精度;③与其他结构件(如插件围框、连接器、插拔器、特殊器件的紧固件、散热片等)的配合方式及尺寸、精度;④为PC板或整机加固而设计的安装孔的尺寸、位置和精度;⑤满足某些特殊要求的屏蔽结构。

在进行外部相关结构设计时要考虑PCB结构与整机结构的协调,如插件与机箱、框架的配合,插件与底板的配合,特别是插件与连接器(插头、插座)的配合,插件与辅助功能结构(如插拔机构)的配合等。此外,还要考虑整机综合性能的要求,可能涉及的设计项目有机械加固设计、整机热设计、电磁兼容性设计、可靠性设计、可维修性设计以及标准化设计等。

内部结构设计的内容为:确定PCB的层数(单面板、双面板、多层板),根据多层板的总厚度确定层间绝缘介质的厚度分配,确定板内各种焊盘、过渡盘、焊接孔、过线孔的尺寸及精度要求等。进行内部结构设计时首先要综合电气性能需求、布线设计要求等各方面因素来确定PCB的层数,应力求减少层数以降低成本和提高可靠性。

对较复杂的电路,特别是器件工作频率较高的电路进行PCB设计时,电源、地线网络的安排对改善馈电特性,调节信号线的传输阻抗,隔离线间干扰,提高布线的布通率等方面都有很大影响。

在设计印制底板时,为了整机运行的安全、可

靠,通常避免在表面层布线。

内部结构所涉及的焊盘、孔等尺寸,可根据有关的标准、规范,结合 PCB 加工的工艺水平来确定。

(2) 布局设计　利用交互或自动的方式合理摆放欲在 PCB 板上安装的元器件。主要过程为:首先在 PCB 板的表面层(顶层或底层)划分布局区域与禁止布局区域,然后用手动方式摆放有特殊要求的元器件,用自动方式摆放其余的元器件,最后根据电特性要求、可测试性要求、布线难易情况等对布局结果进行交互式优化调整。有特殊要求的元器件一般是指安装位置固定的元器件、安装在 PCB 板四周边缘位置上的接插件、需要多种工作电源的元器件和高集成度芯片等。

布局设计的基本原则如下:①满足基本的功能要求。高速总线通常对某些关键信号有特殊的传输延时要求,高速信号处理芯片也常会要求与外围芯片的某些关键连线保持相等的传输延时,所以,相关元器件的摆放必须考虑这些限制,以免布线满足不了要求。②数字信号元器件与模拟信号元器件分区放置,以降低干扰。③有些元器件同时采用 5 V,3 V,1.6 V 的工作电源,应尽量把工作电源相同的元器件靠近摆放,以节省布线空间。④为了有利于元器件之间的互连,应调整元器件间的相邻位置,以提高布通率;元器件的排列方式要有利于通风散热,温升高的元器件放在易于通风散热的部位;元器件的安装焊盘要落在规定的网格坐标点上。⑤易于受干扰的元器件应远离大功率信号器件及大电流信号线,输入输出信号多的元器件应尽量靠近对外互连的接插件,需要隔离、屏蔽的元器件要根据实际需要进行特殊的安排(例如相对集中)。⑥便于测试。⑦便于维修。

(3) 布线设计　利用交互或自动的方式把 PCB 板上应互连的元器件用印制导线连接起来,是印制板设计的核心内容。

布线设计的主要过程如下:在 PCB 板的各层划分布线区域与禁止布线区域;确定设计规则(定义印制导线的宽度、金属焊盘与非金属焊盘的形状与尺寸、过渡孔焊盘的形状与尺寸),确定物理规则或称间距规则(导线与焊盘的间距、焊盘与焊盘的间距、导线之间的间距、导线与金属图形的间距),确定电气规则(定义射极耦合逻辑(ECL)信号、定义差分对、定义信号延时等);首先完成电源信号和地信号的连接,然后用交互或自动的方式完成其余的连线;根据工艺特点对连线进行自动优化处理,比如线平滑、线均匀、45°拐角处理、删除多余过渡孔等;设计丝网印图形;进行设计规则检查,排除错误;完成生产加工文件后处理工作,即分层定义要加工的金属图形、焊盘阻焊图形、元器件的丝网印图形、印制板标记的丝网印图形和钻孔图形,定义 D 码表,最后输出这些图形的工艺文件并提交给生产厂家。

布线设计的基本原则如下:①要保证各个互连线正确、可靠地连通,必要时可采用冗余多点连接。②对于有大电流通过的印制导线,必须考虑要有足够的宽度,以保证其承载电流负荷的能力。③对于有大电位差的相邻印制导线,必须要注意保持适当的间距,防止击穿。④对于数字集成电路,在 PCB 布线时要特别注意减少噪声干扰,基本措施是:利用接地平面(包括电源网络和地线网络)降低印制导线的特性阻抗,避免出现印制导线平行耦合情况或尽力减小印制导线的平行耦合长度,尽可能减少互连引线的长度,对一些关键信号可以附加地线屏蔽。⑤对于线性电路,要特别注意控制反馈,避免产生振荡。⑥对于静电敏感性器件,在设计时要考虑适当的防静电损伤措施。⑦PCB 布线设计要与工程化设计紧密配合,妥善地处理好有关的工程化问题,如电源滤波(电源与地线之间采用分布式的、覆盖足够宽的工作频率的滤波电容),逻辑电路空闲输入端的处理(为防止干扰,将空闲输入端接地或高电平),阻抗匹配,防静电保护等。⑧对一些电容负载能力差的电路,在布线时要注意控制印制导线的分布电容(减少导线的宽度和长度,尽量布在与接地平面距离远的层次上)。

印制板设计原则　印制板设计质量的高低,对计算机的性能,特别是可靠性(参见**可靠性设计**)有重大的影响,甚至影响到计算机的正常功能能否得以实现。要高质量地完成 PCB 设计,必须将计算机整机工程特征、各种元器件的性能特征、PCB 的制造工艺特征、工作环境特征等多方面的相关因素加以有机的综合,在设计过程中统筹考虑,有针对性地处理好有关的工程技术问题,使 PCB 组装的部件和整机的功能、性能得到有效的保障。

印制板设计方法　早期的印制板设计是采用手工绘图的方法进行的。后来,随着计**算机图形学**和**计算机辅助设计**技术的发展,电子产品设计自动化系统(EDA)开始应用(参见**电子设计自动化**),取代了手工设计方法。EDA 可从部件描述开始,按照一定的规则导出逻辑图。设计人员以 EDA 的输出文件、结构设计文件和有关标准规范作为设计输入条

件，利用先进的软件工具，通过人机交互方式完成印制板的外形结构设计、元器件布局、印制板布线、分析与验证、设计规则检查和工艺后处理，最后输出生产加工文件。

20 世纪 90 年代以来，芯片制造技术、表面贴装技术越来越成熟，印制板的密度和板上信号的速度有了极大提高，高速信号电路板的设计成为 PCB 设计的焦点。高速信号电路板通常要解决串扰问题、信号完整性问题、散热问题和电磁兼容问题，未经过分析与验证的高速信号电路板很难保证一次设计的正确性，每一类问题都要依靠专门的软件工具进行分析验证，针对分析出的问题去修改相应的布线或布局。分析与修改的过程可能重复很多次才能最终得到满意的设计结果。 （宫世友 万芙蓉）

yingdaji chengxu sheji

应答集程序设计（answer set programming, ASP） 一种基于约束的，以带应答集语义的逻辑程序为基础的问题解决方法。用 ASP 解决问题时，我们将领域知识表达为用逻辑语言描述的程序（逻辑程序），使得从该程序的应答集中很容易抽取出问题的解。例如，为解决三色问题，我们可以使用谓词 edge(x,y) 表示"图中在节点 x 和 y 之间有一条边"，使用谓词 color(x,y) 表示"节点 x 的颜色为 y"，然后写一个逻辑程序将该问题形式化地表达出来。例如，下面的规则

$$\leftarrow \text{edge}(x,y),\ \text{color}(x,c),\ \text{color}(y,c).$$

表示"不能将相邻节点涂成同一种颜色"。注意，此规则的左部为空，表示一个矛盾。

应答集语义始于 1988 年 Gelfond 和 Lifschitz 为正则逻辑程序提出的稳定模型语义。该语义本质上是命题级的。设 A 为命题符号（原子）的一个集合。正则规则形如：

$$a :- b_1, \cdots, b_k, \text{ not } c_1, \cdots, \text{ not } c_n$$

其中 a 为原子，称为规则的头部；$b_1, \cdots, b_k, c_1, \cdots, c_n$ 均为原子，构成规则的体。正则逻辑程序为一组正则规则。

给定一个正则逻辑程序 P，A 的一个子集 S 称为 P 的一个稳定模型，如果

$$S = \text{consequence}(GL-\text{reduct}(P,S))$$

其中 GL－reduct(P,S) 称为 P 在 S 下的 Gelfond-Lifschitz 约减，它与 P 的区别在于：当且仅当 c 不在 S 中时将 P 中规则体中的"not c"删除。形式化地，它是从 P 和 S 经过下列处理得到的规则的集合：

（1） if P 含有规则，它包含 not c，其中 c 在 S 中，
then 删除该规则。

（2） 删除剩余规则中的 not c 。

显然。GL-reduct(P,S) 是一组确定规则（又称正规则），即规则体中不含"not"。

像 $a :- b_1, \cdots, b_k$ 这样的正规则对应于命题逻辑中的公式 $b_1 \& \cdots \& b_k \Rightarrow a$。设 T 是一组正规则，consequence(T) 就是逻辑上可以从 T 推出的原子的集合。例如，给定下列正规则：

$$a :- b.$$
$$d :- a,\ c.$$

其逻辑推论集合为空：它不能推出任何原子。但若增加规则

b.（规则 b :- 的缩写），我们将得到逻辑推论集 $\{a,b\}$.

现在考虑正则逻辑程序 P，它包括下列规则：

$$a :- \text{not } b.$$
$$b :- \text{not } a.$$

$\{a,b\}$ 共有四个可能的子集。令 $S_1 = \{a\}$。则 GL-reduct(P,S_1) 为：

$$a :-$$

这样，S_1 是 P 的一个稳定模型，因为 consequence (GL(P,S_1)) = $\{a\}$ = S_1。但是，$S_2 = \{\}$ 不是稳定模型，因为 GL-reduct(P,S_2) 为

$$a :-$$
$$b :-$$

它的逻辑推论集为 $\{a,b\}$，与 S_2 不同。类似地，我们可以验证：$S_3 = \{b\}$ 也是一个稳定模型，而 $S_4 = \{a,b\}$ 不是稳定模型。

我们看到，一个程序可能有多于一个的稳定模型。而另外一些程序则可能根本没有稳定模型。例如考虑 $P = \{a :- \text{not } a\}$. $\{\}$ 和 $\{a\}$ 均不是该程序的稳定模型。

规则有时带有变量。例如在三色问题中，我们可以有一条通用规则表示"若一个节点既未涂蓝色，也未涂绿色，则它必定涂了红色"：

color(X,red) :- not color(X,blue), not color(X,green).

此处变量 X 可被图中任一节点代替，即上述规则实为一个规则模式，它代表将 X 替换为图中所有可能节点所得到的一组规则。一般地，给定一组带变量的规则和对象域，将变量替换为该域中图中所有可能对象，我们可以得到的一组无变量的规则。这个过程称为"实例化"。

稳定模型语义后来被推广到包含析取，约束，聚

合和其他特征的更具表达力的逻辑程序，其语义通常被称为应答集语义。

稳定模型语义和应答集语义最初为 Prolog 这类逻辑程序设计语言的“失败为非”算子 not 的形式化而提出的。20 世纪 90 年代后期，Niemela 和他的学生们开发了一个称为 smodels 的系统用于计算带约束的正则逻辑程序的应答集。Smodels 之于逻辑程序如同 SAT 求解器之于命题逻辑。

Smodels 成功地表明，对于像图着色之类的某些组合问题，逻辑程序可以像 SAT 这样的传统求解器一样高效处理。更重要的是，逻辑程序远比命题逻辑的表达力强，它可以自然地表示递归，可以用“失败为非”处理缺省信息。此项工作在有关学术界引起震动。使用逻辑程序和 smodels 这样的计算逻辑程序的应答集（模型）的求解器来解决问题的方法被称为应答集程序设计（ASP）。此后直到目前 ASP 均是一个活跃的研究领域。许多新的 ASP 求解器被设计和实现，包括专门用于析取逻辑程序的求解器 dlv 和一些基于循环公式和可满足性求解器。ASP 已经找到广阔的应用领域，包括产品设计，规划与调度，生物信息学，本体和语义网络。在主要的人工智能国际会议和国际学术刊物上关于 ASP 的论文都占有一席之地。

参考文献

Lifschitz V. What is answer set programming?. AAAI Press, 2008

（林方真）

yingyong fuwuqi

应用服务器（application server） 基于基本的网络通信协议，对网络上各种软硬件资源进行调度和管理，并提供网络应用软件所必需的公共服务和各类构件运行环境的基础中间件。

应用服务器通常以微内核集成总线为基础，由通信服务、公共服务、构件容器、业务引擎和各种工具构成，其概念框架如图 1 所示。

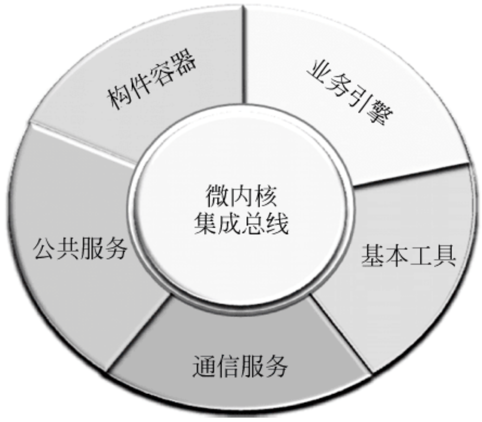


图 1 应用服务器的概念框架

应用服务器各部分的功能是：①微内核集成总线旨在屏蔽底层各种异构的网络和操作系统，在物理位置透明的情况下支持异地构件与服务之间的互访；②通信服务为各种客户提供访问接口和访问协议，如 JRMP，SOAP，IIOP，HTTP/HTTPS 等通信协议，以支持客户对服务器端应用业务逻辑的访问；③公共服务为部署在其上的应用业务逻辑提供企业计算所必需的各种共性基础性服务，如事务、消息、事件、通告、安全、负载平衡、数据访问等，以满足高可用性、安全性、可扩展性和可靠性等企业计算需求；④构件容器提供 EJB，Web，CCM 等实体构件、会话构件、过程构件和服务构件的运行环境；⑤业务引擎用于启动相关构件和服务，如 Portal，BPEL，BPMN 等；⑥基本工具包括各种应用程序的接口定义与相应的编译工具以及各种建模、开发、定制、部署、组装和构件管理工具，以帮助开发者专注于业务逻辑，简化应用开发。

常见的应用服务器有 Java 应用服务器、. Net 应用服务器、PHP 应用服务器等，以 Java 应用服务器最为常见。Java 应用服务器是遵照 Java 企业计算平台规范 J2EE/JavaEE（或缩写为 JEE）实现的，它所定义的 EJB 构件模型清晰地界定了 EJB 构件和 Java 应用服务器中 EJB 构件容器之间的接口，因此，符合标准规范开发的 EJB 构件，能够在不同厂商的 Java 应用服务器上运行。目前国际上主要的 Java 应用服务器商用产品有 IBM WebSphere，Oracle Weblogic，Oracle AS 等，开源产品有 JBoss、JOnAS、Apache Geronimo 等，国内也推出了多款符合相应国际规范的 Java 应用服务器产品，并在我国信息化建设中发挥了重要作用。 （吴泉源）

yingyong ruanjian xitong

应用软件系统（application software system） 直面计算机用户和特定应用领域的**软件系统**。应用软件系统的领域专用性是区别于其他软件系统及**计算机应用技术**的主要特性。

20 世纪 60 年代以前，软件一词尚未出现，应用程序主要用于以数值分析算法为中心的科学计算，持久性的数据量不大，一般由简单的文件系统管理，程序人员通常以单纯手工方式使用机器语言或低级

语言编制程序,有一些数学程序库可供调用。随着实用高级程序设计语言的诞生和发展,60 年代初开始出现软件一词,除科学计算外,出现了一些基于文件系统的事务处理软件,目的是做某些诸如进出账管理和自动生成统计报表等事务性操作。1968 年出现**软件工程**的术语,随着以数据的集中管理和共享为特征的**数据库系统**成为数据管理的主要形式,应用软件系统的重点从求解科学计算问题开始转向数据处理问题,开发方法从手工或个体合作方式逐步转向工程化方法,期间出现的嵌入式应用软件系统在工业控制等领域得到广泛应用。80—90 年代,随着**计算机网络**、个人计算机和软件工程的发展,涌现了大批实用的**管理信息系统**和**办公信息系统**,也出现了一些富有特色的**计算机辅助设计**系统以及与**人工智能**技术结合的**专家系统**。以数据为中心的大规模数据处理在企事业和政府等许多领域得到成功应用。90 年代后期,特别是进入 21 世纪以来,**互联网**的迅速发展,以信息为中心的信息网络服务系统发展为计算机应用的主流。

应用软件系统与应用领域或行业密切相关,且种类繁多,但从技术层面看大致可以分为三类:第一类是科学计算系统,诸如求线性代数方程组、常微分方程组和偏微分方程组等数值解的系统,其处理对象均为数值数据,计算量通常较大,应用领域涉及科学研究、国防建设、工程计算和气象数值预报等。第二类是数据处理系统,诸如**管理信息系统**、**决策支持系统**、办公信息系统、**地理信息系统**等,其处理对象主要还是数值数据,也有一些非数值数据,但计算量一般较小,而数据量和数据传输量较大,应用领域除前述者外,还涉及企业、事业和政府等部门。第三类是信息服务系统,诸如**电子商务系统**,**电子政务系统**,**企业资源规划**系统,**计算机辅助教学系统**,**医疗信息系统**,**数字图书馆**系统,飞机订票系统,即时通信系统,网上购物系统,电子游戏系统,多媒体编辑与播放系统等,其处理对象不仅有数值数据,还包含大量非数值数据和多媒体信息,服务领域几乎覆盖各行各业乃至人们工作和生活的每个角落。以上分类并非绝对,而是互有交叉,且还可以从适用领域或行业以及从管理型、服务型、设计型、控制型和娱乐型等方面进行分类。

应用软件系统的逻辑结构如图 1 所示,通常可以划分为三个层次和一套基础设施:①基础设施　包括支持应用软件开发、部署、运行与管理的计算机硬件或固件、操作系统、网络基础设施和分布式计算环境等。②资源管理层　包括数据库、模型库、知识库和其他信息库,以及实现各类信息资源采集、储存、传输、访问和管理的资源管理系统;对各类结构化、半结构化和非结构化数据实施管理的有**数据库管理系统**、目录服务系统、内容管理系统和元数据管理系统等。③业务逻辑层　由实现各种业务功能、流程、策略和决策等应用业务的一组可执行的信息处理代码组成。④应用表现层　其功能是通过人机交互等方式,将业务逻辑和信息资源紧密结合在一起,并以多媒体等丰富的形式,接受用户输入的信息或命令,向用户发布或展现系统处理的结果。目前网络应用系统的体系结构主要有客户-服务器结构、多级服务器的浏览器-服务器结构和对等服务器结构三种,它们都是上述应用软件系统逻辑结构的实例化。

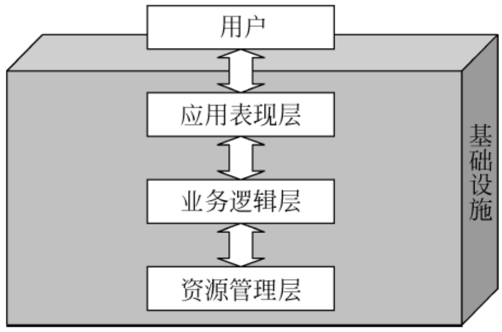


图 1　应用软件系统的逻辑结构图

应用软件系统具有以需求为导向、以计算为核心、以网络为基础、以安全为保障的特点。其开发、运行与维护的方法参见**领域工程**。当前,应用软件系统有如下发展趋势:①结构分布化　应用软件系统的计算任务是分布在网络各结点机上通过信息资源共享完成的;共享的资源不仅有数据,还包括各种计算资源。②系统集成化　构建一个应用软件系统的关键是实现一体化,即把分布在网络各处的各类信息资源实现有机整合和有效管理;集成的内容主要有通信集成、数据集成、代码集成、流程集成和门户集成以及企业对企业或部门对部门之间的服务集成等。③计算智能化　随着信息量的指数级增长和应用软件系统复杂度的不断升级,系统的易用性问题越来越突出,在信息处理中融入各种智能技术成为应用软件系统的发展趋势。主要的智能技术包括**数据挖掘**和知识发现、业务智能和推理引擎、**机器学习**和**自然语言理解**、智能搜索和语义 Web 等。④功能服务化和应用泛在化　随着信息网络科学技术和以软件为突破口的现代服务业的迅速发展,服务构

件化、服务虚拟化和软件即服务的发展潮流日趋显现。一方面,作为一种应用创新模式,应用软件系统将越来越多地以服务或增值服务而不是以产品的形态提供给用户,且用户可以随时随地通过移动无线网等网络设施从应用软件系统中获取所需的服务;另一方面,应用软件系统的开发将越来越多地依赖于由其他软件系统提供的各类基础性服务,这些基础性服务将成为构建应用软件系统的基本单元,应用软件系统开发者只需编写某些组合脚本便可实现一个复杂的业务应用。再者,在分布计算中间件领域化等发展趋势下,应用软件系统所提供的某些重要的特定服务将会沉淀到操作系统或中间件等平台中。因此,应用软件系统的发展丰富了其他软件系统及平台的内涵,其他软件系统及平台的发展也推动着应用软件系统向更复杂应用领域的拓展。

参考文献

Wikipedia. Application software. http://en.wikipedia.org/wiki/Application software　　（吴泉源）

yingshe

映射（mapping）　两个集合元素之间的一种对应规则。映射有时又称函数。

设 X 和 Y 都是集合。若对每个 $x\in X$,根据对应规则 f,都有唯一的一个 $y\in Y$ 相对应,则称 f 为一个从 X 到 Y 的映射或函数,记为 $f: X\to Y$ 或 $X\xrightarrow{f}Y$,并把 y 表示为 $f(x)$,即 $y=f(x)$,这时称 y 为 x 在 f 下的像,x 为 y 关于 f 的原像,并称 X 为 f 的定义域,Y 为 f 的值域空间。

例如,取 $X=\{a,b,c\}$ 及 $Y=\{0,1,2\}$。若对应规则 f 定义如下:

$$a\mapsto 0,\qquad b\mapsto 0,\qquad c\mapsto 2$$

则可得到一个映射 $f: X\to Y$　使 $f(a)=f(b)=0$ 且 $f(c)=2$。

设 $f: X\to Y$。若 $A\subseteq X$ 且 $B\subseteq Y$,则令

$$f(A)=\{y\mid y\in Y \text{且有 } x\in A \text{ 使 } y=f(x)\}$$

$$f^{-1}(B)=\{x\mid x\in X \text{且有 } y\in B \text{ 使 } f(x)=y\}$$

称 $f(A)$ 为 A 在 f 下的像集,$f^{-1}(B)$ 为 B 在 f 下的原像集。并称 $f(X)$ 为 f 的值域,常用 $\mathrm{ran}(f)$ 表示。这时显然有

$$f(f^{-1}(B))\subseteq B\qquad \text{且}\qquad A\subseteq f^{-1}(f(A))$$

而且当 $B\subseteq \mathrm{ran}(f)$ 时,还进一步有 $f(f^{-1}(B))=B$。

设 $f: X\to Y$,$A_1,A_2\subseteq X$ 且 $B_1,B_2\subseteq Y$,则有以下结论:

(1) 若 $A_1\subseteq A_2$,则 $f(A_1)\subseteq f(A_2)$;

(2) 若 $B_1\subseteq B_2$,则 $f^{-1}(B_1)\subseteq f^{-1}(B_2)$;

(3) $f(A_1\cup A_2)=f(A_1)\cup f(A_2)$;

(4) $f(A_1\cap A_2)\subseteq f(A_1)\cap f(A_2)$;

(5) $f^{-1}(B_1\cup B_2)=f^{-1}(B_1)\cup f^{-1}(B_2)$;

(6) $f^{-1}(B_1\cap B_2)=f^{-1}(B_1)\cap f^{-1}(B_2)$。

设 $f: X\to Y$,若 $f(X)=Y$,则称 f 为满射;若当 $x_1,x_2\in X$ 且 $x_1\neq x_2$ 时恒有 $f(x_1)\neq f(x_2)$,则称 f 为内射;若 f 既是满射,又是内射,则称 f 为双射或一一映射。

如果 $f: X\to X$ 使得每个 $x\in X$ 皆有 $f(x)=x$,则称 f 为 X 的恒等映射,用 I_X 表示。

设 $f: X\to Y$ 且 $g: Y\to Z$,这时可定义映射 $h: X\to Z$ 如下:

$$h(x)=g(f(x)),\quad x\in X$$

称 h 为 f 与 g 的合成映射,用 $g\circ f$ 表示。

映射 $f: X\to Y$ 与映射 $f': X'\to Y'$ 相等,是指 $X=X'$,$Y=Y'$,且对每个 $x\in X$ 皆有 $f(x)=f'(x)$。如果 $g: Y\to X$ 使 $g\circ f=I_X$ 且 $f\circ g=I_Y$,则称 g 为 f 的逆映射,并称 f 有逆映射。如果 f 有逆映射,则其逆映射必是唯一的,因此常用 f^{-1} 表示 f 的逆映射。

设 $f: A\to B$,$g: B\to C$ 且 $h: C\to D$,则

$$(h\circ g)\circ f=h\circ(g\circ f)$$

$$f\circ I_A=f=I_B\circ f$$

$$f^{-1}\circ f=I_A\quad \text{且}\quad f\circ f^{-1}=I_B$$

$$(f^{-1})^{-1}=f$$

$$(g\circ f)^{-1}=f^{-1}\circ g^{-1}$$

而且 f 有逆映射当且仅当 f 为双射。　　（王兵山）

yingcipan qudongqi

硬磁盘驱动器（hard disk drive, HDD）　磁盘驱动器的一种,驱动固定安装在主轴上的硬磁盘并实施数据存取的设备,简称硬盘(参见**磁盘存储器**)。硬磁盘是在铝合金或玻璃等硬质圆盘基片表面涂敷磁性薄膜构成。它是目前使用最广泛的,也最能反映磁盘存储技术水平的一种驱动器。世界上第一台硬盘存储器是由 IBM 公司在 1956 年发明的 IBM 360（RAMAC）。它的容量是 5 MB,位密度为 3.9 b/mm(100 bpi),道密度为 0.8 道/mm(24 tpi)。采用固定式盘片,盘片直径是 610 mm(24 in),一共采用了 50 片磁盘,整个驱动器重达一吨,是一个庞然大物。

早期的硬盘通过更换盘片来增加容量,但随着

存储容量的显著增长，通过更换活动磁盘扩大容量的方法已不再必要。1976 年产的 IBM 3350 型固定硬磁盘驱动器问世后，驱动器就以盘片固定的形式发展。盘片固定更有利于记录密度和可靠性的提高。IBM 3350 型磁盘驱动器是在初次采用温切斯特技术(Winchester)的 3340 型驱动器的基础上经过改进而发展起来的。

IBM 3350 及其兼容或同类型产品都采用 356 mm(14 in)或 203 mm(8 in)直径的盘片，整机体积太大，这显然不适合在微型计算机中使用。小型温盘驱动器是利用了大型驱动器成熟的温彻斯特技术根据小型的特点发展起来的。1980 年出现的 ST506，采用直径为 5.25 in 的盘片，虽然容量只有 6 MB，但是体积大小很适合微型计算机使用。ST506 的平均寻道时间为 85 ms，面密度为 1.96 Mb/in^2。20 世纪 80 年代以来，盘片直径从早期的 5 in 向更小的尺寸推移。2003 年，3.5 in 是主流产品。单片容量已经达到 80 GB，单台容量达到 300 GB，转速高达 15 000 r/min。2.5 in、1.8 in 和 1 in 以下的超小型产品也已应用。

硬盘驱动器的技术发展十分迅猛，人们陆续在硬磁盘驱动器中广泛应用了金属薄膜磁层、薄膜磁头或隙含金属磁头(MIG)、微型浮动块、数字伺服以及区位记录(ZBR)技术，还采用了部分响应最大似然(PRML)信号处理技术、磁变阻(MR)磁头和垂直磁记录等新技术，从而在道密度、位密度、存储容量以及数据传输速率方面都有大幅度的提高。特别是由于巨磁阻效应的发现，利用巨磁阻效应的 GMR 磁头读出灵敏度有了巨大的提高，使得硬盘的记录密度又迈上了一个新的台阶。

目前，3.5 in 企业级硬盘容量从 500 GB ~ 3 TB 不等，磁道宽仅为 75 nm，单碟记录密度达到 1 TB，具有 6 Gb/s 的串行嵌入式 SCSI(SAS)和串行 ATA(SATA)接口，平均无故障时间长达 120 万小时，提供了出色的可靠性。

物理结构　硬盘驱动器主要由盘体、控制电路板和接口部件等组成。硬盘的内部结构通常专指盘体的内部结构。如图 1 所示，盘体是一个密封的腔体，里面密封着磁头、盘片及主轴马达、音圈马达等部件。盘片安装在主轴马达的转轴上，在主轴马达的带动下高速旋转。主轴马达采用无刷电机，在高速动压轴承支持下机械磨损很小，可以长时间连续工作。硬盘驱动器采用高精度磁头驱动定位系统，这种系统能使磁头在盘面上快速移动，可在极短的时间内精确地定位到由计算机指令指定的磁道上，高速读取磁盘表面上存储的信息。



图 1　硬盘驱动器的内部结构

硬盘的盘基片是硬质铝合金盘片或者是玻璃盘片，盘厚一般在 0.5 mm 左右，直径主要有 1.8 in(1 in = 25.4mm)、2.5 in、3.5 in 和 5.25 in 4 种，其中 2.5 in和 3.5 in 盘片应用最广。盘片的转速与盘片大小有关，考虑到惯性及盘片的稳定性，盘片越大转速越低。一般来讲，2.5 in 硬盘的转速在 5400 ~ 7200 r/min 之间；3.5 in 硬盘的转速在 4500 ~ 5400 r/min之间。随着技术的进步，现在 2.5 in 硬盘的转速最高已达 15 000 r/min，3.5 in 硬盘的转速最高已达 12 000 r/min。有的硬盘只装一张盘片，有的硬盘则有多张盘片。目前，磁道密度已高达 5400 tpi(每英寸磁道数)或更高；人们还在研究各种新方法，如在盘上挤压(或刻蚀)图形、凹槽和斑点等作为定位和跟踪标记，以提高到和光盘相等的道密度，从而在保持磁盘机高速度、高位密度和高可靠性的优势下，大幅度提高存储容量。

高速旋转的盘体产生明显的陀螺效应，所以，在硬盘工作时不宜搬动，否则，将增加轴承的工作负荷。为了高速存储和读取信息，硬盘驱动器的磁头质量小，惯性也小，所以，硬盘驱动器的寻道速度明显快于软驱和光驱。

磁头　硬盘磁头是硬盘读取数据的关键部件。磁头性能的好坏在很大程度上决定着硬盘的存储密度。早期的磁头是读写合一的电磁感应式磁头，但是，硬盘的读、写却是两种截然不同的操作，为此，这种二合一磁头在设计时必须要同时兼顾到读/写两种特性，从而造成了硬盘设计上的局限。而 MR 磁头(magneto resistive heads)，即磁阻磁头，采用的是分离式的磁头结构：写入磁头仍采用传统的磁感应

磁头(MR 磁头不能进行写操作),读取磁头则采用新型的 MR 磁头,即所谓的感应写、磁阻读。这样,在设计时就可以针对两者的不同特性分别进行优化,以得到最好的读/写性能。另外,MR 磁头是通过阻值变化而不是电流变化去感应信号幅度,因而对信号变化相当敏感,读取数据的准确性也相应提高。而且由于读取的信号幅度与磁道宽度无关,故磁道可以做得很窄,从而提高了盘片密度,达到 200 MB/in,这也是 MR 磁头最初被广泛应用的最主要原因。

目前比较常用的是 GMR(giant magneto resistive)巨磁阻磁头,GMR 磁头使用了磁阻效应更好的材料和多层薄膜结构,这比以前的传统磁头和 MR 磁阻磁头更为敏感,相对的磁场变化能引起大的电阻值变化,从而实现更高的存储密度。磁头是硬盘中对盘片进行读写工作的工具,是硬盘中最精密的部位之一。磁头是用线圈缠绕在磁芯上制成的。硬盘在工作时,磁头通过感应旋转的盘片上磁场的变化来读取数据,通过改变盘片上的磁场来写入数据。为避免磁头和盘片的磨损和碰撞,在工作状态时,磁头悬浮在高速转动的盘片上方,而不与盘片直接接触,只有在电源关闭之后,磁头会自动回到在盘片上的固定位置(称为着陆区,此处盘片并不存储数据,是盘片的起始位置)。

温切斯特技术 当前硬盘普遍采用了温切斯特(Winchester)技术,其基本特点如下:

(1) 磁头、磁盘片、主轴及装载头臂的小车等机械公差要求严格的关键零部件装在一密封的壳体内,此部件称作头盘组合件(HDA)。

(2) 低质量、小尺寸、轻浮力的平轨磁头浮动块,可与盘片作接触启停,盘片磁层表面涂有润滑剂,盘片不转时头与盘接触,当盘达到一定转速时浮动块浮起,当盘速降至一定值时浮动块经一段滑行后降落到盘面上。

(3) 读前置放大电路、写电流驱动电路及选头电子开关均集成在一个芯片上,此芯片安装在各磁头存取臂上以便减少外界电磁场对磁头引线的影响。

采用温切斯特技术时,HDA 安装在驱动器的减震架上,需和驱动器连接的部位仅限于主轴电机、存取臂驱动音圈电机、净化空气输入口及读写电路接插件等处。由于磁头和磁盘组合成为一个整体,磁头与磁盘的相对关系是固定的,磁头所读出的乃是由它本身写入的信息,因此无须像可换盘驱动器那样为保证盘片的可换性而考虑严格的尺寸公差要求,从而使密度得以大幅提高。以 IBM 3350 型驱动器的推出为转折点,固定硬磁盘驱动器一直持续向前发展。

(刘锡刚　周功业)

yingcipan qudongqi ceshi

硬磁盘驱动器测试(hard disk drive testing)

通过测试仪对硬磁盘驱动器机械、电气性能进行的检测和评估。硬磁盘驱动器应在研制、生产、维修等过程中,进行严格的检测和评估(对于可换盘硬磁盘驱动器还应检测其互换性),以保证其质量。

硬磁盘驱动器机械与电路的测试和调整内容主要有:索引信号,零道信号,准备好信号,主轴转速,寻道,磁头定位(径向位置及方位角),磁头分辨率,磁头读出信号幅度,磁头写入波形,数据信号抖动性,内、中、外磁道信号的不对称度(测信号飘移),内、中、外磁道读裕量(测误码率),连续写读,格式化磁盘。

硬磁盘驱动器测试仪大致可分练习仪和测试分析仪两大类。

(1) 练习仪　是一种低档的非智能或具有一定编程功能的简易型检测仪。一般用于完成对硬磁盘驱动器的索引信号、零道信号、准备好信号、磁头寻道、写读、选头等功能测试,与示波器联用时还可观测驱动器读写数据信号。检测和调整可换盘硬磁盘驱动器时,还要用到调整盘。练习仪结构简单,轻巧,使用方便,一般供维修使用,每次仅可测试一台驱动器。

(2) 测试分析仪　是一类中高档功能检测仪,采用微处理机或计算机控制,可自动操作,也可手动操作。除了练习仪的基本功能外,还能对硬磁盘驱动器进行读信号、相位裕量、数据分离、信号不对称度等指标的定量分析,可用来对硬磁盘驱动器进行全面的检测和评估。这类仪器能精确地进行出错定位和记录,并可同时或顺序测试多到 64 台驱动器。测试程序、测量值及测试结果可在显示屏上实时显示或同时由打印机打印输出。仪器的结构形式有便携式、台式、落地式等。

参考文献

Greenwood D. Disk-drive testers. Electronics Test, 1984, (2): 68-81

(周学仁)

yingjian miaoshu yuyan

硬件描述语言(hardware description language,HDL)　描述电子系统(特别是数字系统)

硬件结构与功能行为的语言。硬件描述语言既可用于表达门级、芯片级、插件及系统级的硬件设计，又是一种统一的设计数据格式和信息交换标准。应用硬件描述语言，能够进行逻辑设计、逻辑综合、模拟验证、测试、诊断和修改。

硬件描述语言的研究始于20世纪60年代，如1965年发表的CDL，1968年发表的DDL，1973年发表的AHPL等。这些语言之间实质性的差别很少，但因为没有统一和推广，妨碍了它们的发展。第一个真正有影响的硬件描述语言是1983年Gateway Design Automation公司（1989年被Cadence公司收购）的Philip R. Moorby设计的Verilog语言。当时该语言只是用于数字系统功能行为的验证和模拟，并不能代替原理图输入。基于Verilog语言开发出的Verilog-XL至今仍是HDL模拟器中的经典。后来逻辑综合技术逐步成熟，由Verilog语言描述的RTL电路设计经逻辑综合工具变换为门级网表后，相比原理图输入，并无太多的效率损失，Verilog才正式走上集成电路设计的主战场，被工业界的绝大部分芯片设计项目所采用，并于1995年成为IEEE 1364号标准。

超高速集成电路硬件描述语言VHDL的发展与Verilog正好相反，它是先标准化，再逐步得到应用的。1980年美国国防部提出超高速集成电路（VHSIC）发展计划，要求各承担超大规模集成电路产品订货的公司都要以新设计的硬件描述语言VHDL作为统一的设计数据文件格式和信息交换工具，并用它的模拟系统模拟验证。1986年IEEE以VHDL 7.2版本为基础制定了标准的VHDL版（IEEE 1076）。

无论是Verilog还是VHDL，都有下列共同的特点：

（1）能描述硬件的构成，即微电子器件、元件、系统的功能构成；

（2）能描述硬件的行为，即微电子器件、元件、系统如何工作，包括数据转换、定时、电信号的变化等；

（3）能描述设计实体，说明元件的外部特性，以便组装引用；

（4）能描述结构的连接、组装关系，以便构成一个复杂的系统；

（5）具有丰富的类型，除具有数值类型、字符类型、枚举类型外，还能表示时间、电容量、阻值等物理量；

（6）具有丰富的表达能力，除提供一般的说明、算术、逻辑运算等过程型语句外，还提供相当强的并发执行能力；

（7）得到了主流**电子设计自动化**（EDA）厂商在模拟验证、逻辑综合等相关工具方面的支持；

（8）主要支持数字电路设计，对模拟电路的支持比较弱。

与普通的程序设计语言相比，硬件描述语言有两个最主要的特点：①引入了时间的概念，例如，逻辑操作具有时延特性、时序部件只在时钟周期中的某个时刻翻转等；②具备并发执行的能力，即允许电路中的许多不同部分各自独立地向前推进。所以通常情况下，硬件描述语言和**程序设计语言**在功能上是不能互相替代的。随着复杂**片上系统**（SoC）产品对硬软件协同设计和协同验证需求的不断增强，业界开始研究能同时支持硬件设计和软件开发的高层次语言及相应的设计验证方法和工具。SystemC和SystemVerilog的出现就是这类尝试的初步结果。另一种有益的尝试是在硬件描述语言中增加对模拟电路的支持，以增强模拟和混合集成电路的自动设计能力。

参考文献

1. Thomas D, Moorby P. 硬件描述语言Verilog. 4版. 刘明业，等译. 北京：清华大学出版社，2001

2. 王志华，等. 数字集成系统的结构化设计与高层次综合. 北京：清华大学出版社，2000

（潘雪增　唐志敏）

yingjian tongbu jizhi

硬件同步机制（hardware synchronization mechanism）　在**并行处理系统**中，通过采用硬件实现低层或原语级控制信息的通信，以保证多个进程具有正确执行顺序的机制。同步的目的是保证并行处理结果的正确性，它可以通过硬件、固件或软件（用户程序或操作系统）实现，硬件同步机制是任务层软件同步的基础。同步是获得并行加速所付出的主要开销，有效的硬件同步机制是实现并行处理的一项关键技术。

在单处理机中，指令的顺序由程序计数器确定。早期的计算机未提供不可**中断**的读改写指令，实现同步非常困难。E. W. Dijkstra早在20世纪60年代就提出了只用ALGOL 60语言的标准语句实现同步的方案，他的方案中实际上隐含了多处理机必须保持顺序一致性这一假定，即并行处理系统中所有处理机观察到的各处理机的存取操作都是按同一次序

进行的。由于存取数据可以在寄存器、**高速缓存存储器**、**主存储器**以及**磁盘存储器**中进行,存取时间相差很大,先执行的存取指令可能后完成。从别的处理机观察一个处理机,观察到的存取次序可能与这个处理机上串行编程规定的次序是不一致的。如果要求多处理机系统运行时严格遵守顺序一致性,就可能导致系统性能的明显下降。对于当今的多处理机系统,E. W. Dijkstra 的同步方案已不重要,因为功能不断增强的不同形式的读改写硬件同步原语已逐渐被采用,专门的同步指令以及比顺序一致性要求低的各种一致性协议可以保证并行计算结果的正确性。

硬件同步机制的基础是不可中断的原子操作,所谓*原子操作*是指执行读改写方式的同步指令时,必须等指令执行完毕,别的指令才能执行,执行过程不允许中断。为了防止同时修改一个共享变量,保证数据的完整性与顺序地进入临界区,同步原语必须具有原子性。所谓临界区是指必须顺序执行的一段程序,每个临界区由一个同步变量即信号量保证每一时刻最多只有一个进程进入。已被采用的硬件同步原语或指令有好几种,其中具有代表性的包括测试与设置,增 1 与减 1,比较与交换,取与加等。

(1) 测试与设置(Test and Set)是最简单也是最早采用的硬件同步原语。使用这一原语要求每一共享数据附加一位专门的同步变量,或称信号量。存取共享变量时先要执行一条不可中断的读改写方式的指令 Test and Set。第一个通过 Test and Set 同步原语发现信号量为 0 的进程被允许存取共享变量,在执行 Test and Set 原语时已将信号量置 1。因此,在一个进程存取共享变量时,其他进程不可能再存取共享变量,从而保证了互斥存取。当存取结束时再将同步变量置 0,允许其他进程通过竞争去存取共享变量。

Full/Empty 同步原语与 Test and Set 类似,只是增加了两个两位字段 SAC 和 DAC,分别控制同步读与同步写,而且共享变量与信号量可同时存取,以节省存取时间。这一机制也已用于数据流计算机,以保证正确的读写次序。

(2) 增 1 与减 1(Increment and Decrement)可以看成是 Test and Set 原语的扩展。考虑一个长度为 M 的共享缓冲器,即缓冲器含有 M 个共享的存储单元。初始时共享缓冲器的信号量置为 M。只要不同进程访问缓冲器中的不同单元,最多 M 个进程可同时访问缓冲器。每一进程进入临界区,信号量减 1,完成临界区操作则加 1。Test and Set 原语相当于缓冲区长度为 1 的情形,每一时刻最多只允许 1 个进程进入临界区。Increment and Decrement 原语比 Test and Set 原语效率高,但如果处理不当,可能出现*活锁*,即当大量进程同时要求访问缓冲器时,可能使信号量小于 0,使得一段时间内无法对缓冲器进行存取。

(3) 比较与交换(Compare and Swap)硬件同步原语将临界区缩小到只有 1 条比较与交换指令,这明显地降低了同步开销。执行 Compare and Swap 需要两个寄存器,一个存共享变量旧值,另一个存其新值。计算共享变量新值时不能进入临界区,当新值算出结果以后,再执行不可中断的 Compare and Swap 指令,以测试共享变量的值是否改变,如未改变,则赋以新值,如已改变,则将共享变量当前值存于旧值寄存器。与前两种同步原语不同,在执行 Compare and Swap 之前,允许多个处理机修改共享变量。队列指针是最常用的共享变量,Compare and Swap 原语提供一种途径,允许多个处理机同时修改队列指针而只花费很小的开销。Compare and Swap 是一种很有效的同步原语,但正确地使用这一原语较困难。

(4) 取与加(Fetch and Add)是一种完全并行、不需要临界区的同步原语,多台处理机可同时执行 Fetch and Add 指令。执行 Fetch and Add 指令要求**互连网络**具有存储和整数加法功能。当多个处理机通过 Fetch and Add(V,e)指令存取同一共享变量 V 时,通过合并网络的开关模块,将存取及修改(即增量 e_i)要求合并成一个包括总的增量 e 的存取要求。不管有多少个同时存取的要求,只需对共享变量读改写 1 次。从存储器中取到的结果再通过网络开关按每个处理机不同的修改要求送到各处理机。Fetch and Add 同步原语取消了必须顺序执行的临界区,大大降低了同步开销,为构造例如 1000 到 10 000个处理机组成的大规模并行处理系统提供了一种高性能的同步机制,但由于附加的硬件成本太高,编程也十分困难,不适于中小规模的并行处理系统。

在**消息传递**类型的并行计算机中,消息的发送与接收起到与共享存储计算机中同步原语一样的同步作用,对消息传递同步机制最基本的硬件支持是连接发送与接收处理机的通信通道。消息传递的同步原语非常简单,可以用软件通信协议或专门的硬

件实现。通信可以是同步方式,也可以采用准异步方式或完全异步方式。在完全异步方式的消息传递中,接收方不发出任何请求,发送方只要数据准备好就发送,不管接收方当时是否需要。这种同步原语对程序员完全透明,编程非常方便,但接收方必须有足够容量的匹配单元随时接收发来的消息。

设计低成本高性能的硬件机制支持并行计算机的同步一直是计算机追求的目标,采用硬件实现的屏障是受到广泛注意的同步技术。采用加法器、与门、线或等简单器件可以实现屏障功能。对于包含上千个处理机的并行处理系统可以采用总线连接若干专门的屏障芯片实现同步。数据流计算机的每一条指令都要同步,高效率的同步机制显得更为重要,需要特殊设计的匹配单元实现同步。

实现**高速缓冲存储器一致性**也能起到同步的作用,对于规模较小的系统,用基于总线的高速缓冲存储器一致性协议实现同步可能是合适的选择,但对于包含数百个处理机以上的并行处理系统,专门的同步子系统可能更合算。比如,Cray T3D 系统实现了能够支持数千个处理机的"屏障"和"找到"等同步操作的专用硬件子系统。并行处理系统的同步性能可用每秒执行多少百万条同步指令来表示,记为 MSYPS。一般来讲,MSYPS 远小于 MIPS(百万条指令每秒)。增加处理机数量有可能提高系统的 MIPS,但不能增加 MSYPS。统计表示,平均每执行 50 条左右的指令就要执行 1 条同步指令,因此,MSYPS 是影响并行处理系统性能最主要的因素,应当受到重视。

参考文献

1. Stone H S. High performance computer architecture. Reading, MA: Addison Wesley, 1993

2. 黄铠,徐志伟. 可扩展并行计算:技术、结构与编程. 陆鑫达、曾国荪、邓倩妮,等译. 北京:机械工业出版社,2000

(李国杰)

yinglianxian kongzhiqi

硬连线控制器(hard-wired control unit) 通过基本逻辑电路生成实现机器指令功能所必需的各种基本操作的控制器。相对于**微程序控制器**,其主要特点是各种操作控制命令的逻辑表达式是由与门、或门、非门、与或门等基本逻辑电路完成。又称组合逻辑控制器。

硬连线控制器由指令部件、地址部件、时序部件、操作控制部件和中断控制部件等组成(参见**控制器**)。其中操作控制部件用来产生各种操作控制命令,它根据指令要求和指令流程,按照一定顺序发出各种控制命令。操作控制部件的输入信号有指令译码器的输出、时序信号和运算结果标志状态信号等。设计时根据指令流程、操作时间表得到各种操作控制命令的逻辑表达式,可采用基本逻辑电路与门、或门、与非门等组成的逻辑网络来实现,也可采用可编程逻辑器件 PLD 来实现。PLD 的"与"阵列及"或"阵列和操作控制命令的"与-或"逻辑表达式相对应,为设计组合逻辑控制器提供了一种理想器件。20 世纪 80 年代出现的通用阵列逻辑电路 GAL 与 PAL(参见**专用集成电路**)具有类似的结构,它不但可编程并且是可擦除的,提供了更大的灵活性。

硬连线控制器的最大优点是速度快。但因其线路复杂而且不规整,不便于调试、维护、修改,也不便于仿真不同的机器指令集。

参考文献

金兰,金波. 计算机组织:原理、分析与设计. 北京:清华大学出版社,2006

(谢树煜)

yongsai kongzhi

拥塞控制(congestion control) 拥塞是分组交换网络的伴生现象。当输入网络的流量超过其固有容量时,不可避免地产生拥塞。拥塞的直接结果是分组丢失率提高,端到端延时增加,甚至有可能导致整个网络传输系统的崩溃。拥塞控制是网络流量管理的基础技术之一,主要通过监测或推断网络状态,并在局部或全局针对性地调节分组转发或传输的速率,在防止过量业务流量导致拥塞崩溃的前提下,尽可能提高资源利用率。拥塞控制机制有两种基本形式,即开环拥塞控制和闭环拥塞控制。

开环拥塞控制不依赖于任何网络状态反馈信息,端系统或网络中间节点根据预先设定的配置参数,参照局部信息(如链路带宽或缓存队长等),对流量进行监测与控制,或对分组进行标记或丢弃。依据作用点的不同,又可将开环拥塞控制分为端系统策略和网络节点策略。前者多表现为端系统上的流量速率控制,如 IP 网络差分服务模型中的流量监测(traffic policing)和 ATM 网络中的用法参数控制(usage parameter control)等;网络节点策略多表现为与分组调度和队列管理相关联的分组丢弃策略。一般而言,开环拥塞控制较难适应网络状态的动态变化,鲁棒性差。

闭环拥塞控制一般指依赖网络状态反馈信息的

流量控制策略。反馈既可以是显式的,也可以是隐式的。显式反馈必须发送独立的分组承载状态信息,有时也采用搭载(piggyback)方式。显式反馈策略又可进一步分为持久性反馈和响应性反馈。持久性反馈指以确定的周期产生反馈,响应性反馈指仅在一定条件下才被触发反馈。隐式反馈方式则无须发送独立消息,而是从相关网络事件中推测网络状态信息,如分组丢弃、确认分组的延时、分组到达速率等。流量控制有逐跳(hop-by-hop)和端到端(end-to-end)两种基本模式。逐跳(hop-by-hop)模式下,与会话连接相关联的每一跳链路独立缓存和控制分组发送,利于实现分组零丢失。InfiniBand 信誉流量机制和无线自组织网络多采用这种模式。端到端流量控制通过反馈信息探测与会话连接相关联链路的最小可用带宽,端系统采用加性增加倍乘减小等调节机制使注入网络的流量接近最小可用带宽,实现拥塞避免与拥塞恢复。网络节点则尽可能快地转发业务分组,不施加其他调控措施。端系统中的流量调节一般采用基于窗口或基于速率的基本方式来实现。IP 网络的 TCP 流量控制采用窗口机制,而 ATM 网络自适应比特业务流量控制与增强型以太网链路层的流量控制则采用速率机制。

参考文献

1. Kourose JF, Rose KW. 计算机网络——自顶向下方法. 4 版. 陈鸣,译. 北京:机械工业出版社, 2009.

2. Yang C-Q, Reddy AVS. A taxonomy for congestion control algorithm in packet switch networks. IEEE Network Magazine, 1995, 9(4): 34-45

3. Van Jacobson. Congestion avoidance and control. ACM SIGCOMM, 1988: 314-329 (任丰原)

yonghu jiemian

用户界面(user interface) 计算机系统中实现用户与计算机通信的软硬件设施。又称用户接口,或人机界面。

用户界面的硬件部分包括用户向计算机输入数据或命令的输入装置及由计算机输出供用户观察或处理的输出装置。用户界面的软件部分包括用户与计算机相互通信的协议、约定、操纵命令及其处理软件。目前常用的输入输出装置有键盘、鼠标器、显示器、打印机等。常用的人机通信方法有命令语言、选项、表格填充及直接操纵等。

从计算机用户界面的发展过程来看,可分若干阶段:

(1) 控制面板式用户界面 这是计算机发展早期,用户通过控制台开关、板键或穿孔纸带向计算机送入命令或数据,而计算机通过指示灯及打印机输出运行情况或结果。这种界面的特点是人去适应现在看来是十分笨拙的计算机。

(2) 字符显示式用户界面 分时操作系统出现,使多个用户通过交互显示终端同时分享计算机资源。用户通过键盘输入字符型数据或命令,通过显示终端观察字符型的数据输出。这种界面的通信方法主要采用作业控制语言,后来发展为命令语言。其优点是功能强、灵活性好、屏幕开销少等。缺点是难学、难记、易错、需要一定键盘操作技能、显示不直观等。

(3) 图形用户界面 在 20 世纪 80 年代计算机图形技术基础上发展起来的图形用户界面,是当今计算机用户界面的主流。它采用 WIMP 技术,即窗口(window)、图符(icon)、选单(menu)及指点设备(pointing device)技术,采用多窗口系统为主要软件,以直接操纵为主要使用方法。其优点是操作简便、显示直观形象、适合初学者等。其缺点是对硬件资源要求较高、软件实现较复杂等。图形用户界面仍在不断改进发展中,如增加网络浏览功能、增强多媒体和三维功能、支持应用数据可视化、开发更好的界面构造工具和语言等。

(4) 新一代用户界面 虚拟现实技术将用户界面的发展推向一个新阶段:人将作为参与者,以自然的方式和计算机生成的虚拟环境进行通信。以用户为中心、自然、高效、高带宽、非精确、无地点限制等是新一代用户界面的特征。多媒体、多通道及智能化是新一代用户界面的技术支持。语音、自然语言、手势、头部跟踪、表情、视线跟踪等新的、更加自然的交互技术,将为用户提供更方便的输入技术。计算机将通过多感知通道来理解用户的意图,实现用户的要求;计算机不仅以二维屏幕向用户输出,而且以真实感(立体视觉、听觉、嗅觉、触觉……)的计算机仿真环境向用户提供真实的体验。新一代用户界面是计算机科学技术中最富有挑战性的领域之一。

参考文献

1. Helander M. Handbook of human-computer interaction. Elsevier Science Pub. (North-Holland), 1988

2. Foley J D, VanDam A. 交互式计算机图形学基础. 唐泽圣,周嘉玉,译. 北京:清华大学出版社, 1986

3. 董士海. 计算机用户界面及其工具. 北京:

科学出版社，1994

4. 丁茂顺. 用户接口技术与交互系统构造方法. 北京：科学出版社，1992　（董士海　蔡士杰）

yonghu jiemian guanli xitong

用户界面管理系统（user interface management system，UIMS）　支持设计、构造、执行、评价、维护及管理计算机用户界面的软件。

随着人机交互系统的发展，用户界面已成为计算机系统的重要组成部分，它影响整个系统的可用性及使用效率。开发高质量用户界面需要很大工作量，因此人们希望把它从计算机系统中分离出来。以便使用户界面的开发不受系统功能核心的设计或改动的影响。这种界面与系统功能核心"分离"的思想促使了用户界面管理系统的诞生。W. Newman 在 1968 年建立的"反馈处理器"是最早的 UIMS。但 UIMS 作为专门术语是 D. J. Kasik 在 1982 年首次提出的。早期 UIMS 的功能仅限于原型构造或显示管理，可用性较差。1982 年召开的图形输入交互技术（GIIT）研讨会首次阐述了 UIMS 的概念、作用及结构模型等，此后 UIMS 有较快发展，出现了 TIGER，GWUIMS，Alberta UIMS 等实验系统。20 世纪 80 年代中后期出现了一些商品化 UIMS。我国也研制了一些实验性系统。

用户界面管理系统由两部分组成。其一是构造用户界面的工具集，称为用户界面开发环境或设计环境（UIDE），其二是用户界面运行支持系统，称为用户界面系统（UIS），它与应用系统的功能核心共同组成该应用系统。UIMS 框图如图 1 所示。

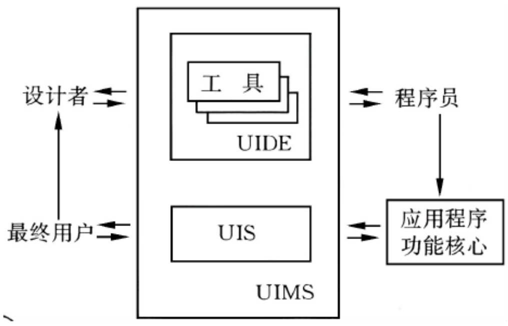


图 1　UIMS 框图

目前多数 UIMS 系统中 UIDE 的结构组成采用了 1983 年在 Seeheim 举行的用户界面管理系统国际研讨会上提出的 **Seeheim 模型**（如图 2 所示）。图中表示部件负责用户界面的外部表现，如屏幕管

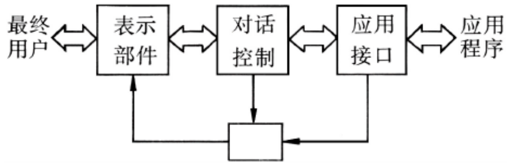
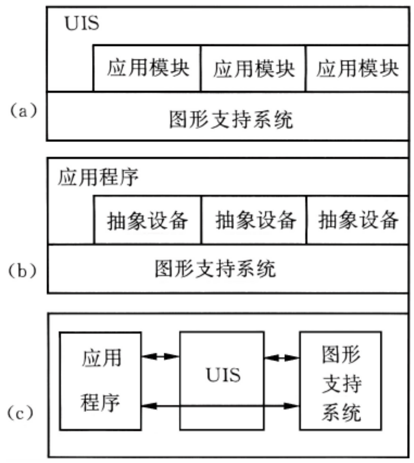


图 2　Seeheim 结构模型

理、图形生成、设备管理、词法反馈及将输入数据转换成内部格式等。对话控制部件将表示部件的数据或请求，经检验传送给应用程序中合适的接口例程，它也将应用程序对请求的回答或其他数据传给表示部件。应用接口是从用户观点看到的应用程序的一种表述，它包括有关数据结构或对象的描述。用户可使用的有关例程（语义）及其限制等。由于 Seeheim 模型只是一个"执行"模型，不反映整个界面软件生存期，且对于直接操纵方式所需的语义反馈很难适应，目前已提出许多改进。

UIMS 系统中 UIS 与应用系统功能核心的关系，涉及用户界面的运行控制方式，它可分为三种类型：外部控制、内部控制及并行控制，分别如图 3 的（a），（b），（c）所示。外部控制是指用户通过 UIS 进行输入，并以此激发应用程序的各功能模块。内部控制是指应用程序根据输入请求的不同，来取得相应抽象设备的支持。并行控制是指应用程序功能核心与用户界面系统在同一层次上并行运行，它们间的通信协调通过并发进程来实现。



（a）外部控制；（b）内部控制；（c）并行控制

图 3　界面的三种运行控制方式

UIMS 的发展趋向是提高实用性，支持直接操纵方式，开发支持复杂用户界面（如三维图形界面、多媒体用户界面、分布式用户界面）的设计技术，进一步重视人的因素研究等。尽管目前 UIMS 在可用性

及功能方面还有不足,但它在提高界面开发效率及质量方面已显示极大潜力,期望它将会像现在的数据库管理系统一样广泛使用。

参考文献

1. 董士海. 计算机用户界面及其工具. 北京:科学出版社,1994

2. 程景云,等. 人机界面设计与开发工具. 北京:电子工业出版社,1994

3. Pfaff G E. User interface management. Springer -Verlag, 1985

(董士海)

yonghu jiemian xiaolü pingjia

用户界面效率评价(evaluation of user interface efficiency) 在人机交互领域中对人机界面的有效性进行定量评价的方法。随着信息技术的发展,人们生活中无时无刻不在与计算机发生着各种形式的交互。人机交互界面,以硬件或者软件的形式,承载着人机间的信息交换任务。定量的分析、评价人机界面作为人机间信息传输的通道的效率,是指导设计高效的人机界面的基础。

对人机界面效率的评价主要集中在输入效率方面。由于影响人机界面效率的因素涉及软、硬件和人因等多方面的影响,研究人员在不同层次上开展了对界面效率评价的研究。GOMS(goals, operators, methods and selection rules)模型抽象了人机交互过程中的四个交互要素,包括目标(goals)、操作(operators)、方法(methods)和选择规则(selection rules)等,并将为达到某一目标,按照一定的方法所采取的交互操作顺序地形式化表达出来,应用选择规则计算各个基本操作(包括击键、指点、归位、心理准备等)的平均用时,得到用户在交互界面上完成特定任务所需要的时间,并以此时间作为评价界面效率的依据。Fitts' Law 则从生理学角度出发,定量建立了指示器从当前位置到达目标的时间同移动距离以及目标大小的关系,已成为 GUI(graphical user interface)中指点操作时间的主要度量方法,指导着各种 GUI 交互界面的设计。Fitts' Law 表示为:

$$T = a + b\log_2(D/S + 1)$$

其中,D 是当前指示器到目标的距离,S 是目标的大小,a, b 为常数,T 为移动指示器从当前位置到目标位置的时间。

相关的研究还包括 Steering Law 和 Hick' Law,前者用于预测用户控制指示器沿着二维平面上的路径移动所需要的时间,可以指导笔式交互界面的设计,而后者则可用来估计用户进行菜单选择的时间开销。

人机界面技术本身仍在发展中。形式多样的交互界面在不断涌现,比如触控界面、笔式交互界面和语音界面等。由于不同的界面系统有着各自的特点和影响效率的因素,这就需要研究人员有针对性地开展界面效率评价的研究,提出面向不同界面系统的、细化的评价方法。同时,人机界面正朝着个性化的方向发展。因此,提出指导个性化界面设计的界面效率评价方法,使之更好地适应于个体的交互任务,也是界面效率评价研究的方向。

参考文献

1. Card S K, Moran T P, Newell A. The psychology of human-computer interaction. Hillsdale, NJ: Lawrence Erlbaum Associates Inc., 1983

2. Accot J, Zhai S M. Beyond Fitts' law: models for trajectory-based HCI tasks. Proceedings of ACM CHI Conference on Human Factors in Computing Systems, 1997

(史元春)

yonghu shujubao xieyi

用户数据报协议(user datagram protocol, UDP) Internet 上一种提供应用程序之间传送数据报的协议。它提供不可靠的无连接数据报传送服务,适用于对传输速率要求较高、通信量比较大,而对于可靠性和安全的要求较低的场合。

每个 UDP 报文称为一个用户数据报,分 UDP 报头和 UDP 数据区两部分。UDP 位于 IP 层之上。应用程序访问 UDP 层,然后使用 IP 层传送数据报。将 UDP 层放到 IP 层之上,表示一个 UDP 报文在互联网中传输时要封装到 IP 数据报中。最后,网络接口层将数据报封装到一个帧中再进行物理传输通道上的传输。

UDP 协议的每个数据报根据其目的主机的 IP 地址独立进行互联网中的**路由选择**。为了在给定的主机上能识别多个目的地址,同时允许多个应用程序在同一台主机上工作并能独立地进行数据报的发送和接收,UDP 将每台机器看作是一些抽象的协议端口的集合,通过协议端口能区分在一台机器上运行的多个程序。每个 UDP 报文不仅传送用户数据,还包括发送方和接收方的协议端口号,以使接收方的 UDP 软件能将报文送到正确的接收进程,并回送应答报文给对应的发送进程。

UDP 使用底层的**网际协议**(IP)来传送报文,因

此它不具备报文到达确认、排序以及流量控制等功能,报文可能会丢失、重复以及乱序等。而可靠性的问题将由使用 UDP 的应用程序自己来解决。

IP 层的报头指明了源主机和目的主机的地址,而 UDP 层的报头指明了主机上的源端口和目的端口。

UDP 也提供复用和分解的功能。它接收多个应用程序送来的数据报,把它们送给 IP 层去传输,同时它接收 IP 层送来的 UDP 数据报,把它们送给对应的应用程序。

从概念上讲,所有的 UDP 软件与应用程序之间的复用和分解都要通过端口机制来实现。实际上每个应用程序在发送数据报之前必须与操作系统进行协商以获得协议端口和相应的端口号。凡是利用指定的端口发送数据报的应用程序都要把端口号放入 UDP 报文中的源端口字段中。

UDP 端口号的指定有两种方式:一种是由某些管理机构指定的,称为著名端口,供用户使用;另一种是动态绑定方式,由应用程序指定端口。

参考文献

1. 胡道元. 计算机网络. 2 版. 北京:清华大学出版社, 2009

2. Comer D E. Internetworking with TCP/IP, Volume Ⅰ. 4th ed. Englewood Cliffs, NJ: Prentice Hall, 2000

（胡道元）

yonghu wangluo jiekou

用户网络接口(user network interface, UNI)　用户终端设备和接入网(AN)之间的接口,用户终端通过 UNI 连到 AN。AN 通过 UNI 为用户提供各种业务服务。UNI 进一步分为独享式 UNI 和共享式 UNI。所谓独享式 UNI,是指用户终端设备(TE)通过 UNI 只能接入一个 SN;所谓共享式 UNI,是指用户终端设备(TE)通过 UNI 可以接入到多个业务节点(SN)。

独享式 UNI 又称为单个 UNI。在单个 UNI 情况下,逻辑用户端口功能和 UNI 的传输媒体层终端被看成是一组相容的功能组。AN 中使用用户网络接口来支持当前提供的接入类型和业务,包括各种公共电话交换网络(PSTN)和综合业务数字网(ISDN)的 UNI。但是 PSTN 中的 UNI 和用户信令并没有得到广泛应用,因而通常各个国家采用自己的规定。

共享 UNI 的例子是异步传送模式(ATM)接口。当 UNI 是 ATM 接口时,这个 UNI 可支持多个逻辑接入,每一个逻辑接入通过一个 SNI 连接到不同的 SN。这样,ATM 接口就成为一个共享 UNI,通过这个共享 UNI 可以接入多个 SN。

参考文献

1. 通信行业标准:VB5. 1 接口技术规范. YD/T 997—1999. 1999

2. 通信行业标准:数据通信名词术语. YD/T 1133—2001. 2001

（程时端　马严）

youxianyuan fangfa

有限元方法(finite element method)　求解二维或三维连续体内偏微分方程(特别是椭圆型方程)边值问题的一种数值计算方法。该法的基础为变分原理和剖分插值,前者是把需求解的偏微分方程化为等阶的泛函极小问题,后者则是把求解区域的连续体剖分成有限多个小的单元体,并用简单的分片插值函数来代替所求的未知函数,这两者相结合,即把一个在连续体区域上求解偏微分方程的问题化成为在区域内部的和边界的一组离散点集上求解代数方程组问题,这种离散化方法称之为有限元方法。它与传统的有限差分离散方法比较有以下显著的优点:①适合于任意复杂的求解区域以及区域内不同介质的处理。②边界条件处理方便自然。③离散方法统一,特别适合于大型电子计算机编制程序,现已研制出许多通用的有限元分析软件包提供用户使用。

变分原理　许多椭圆型方程边值问题,都与一个适当的“泛函极小原理”(变分原理)相等价,这样就把一个求解偏微分方程问题化为一个求解泛函极小问题。如二维弹性力学偏微分方程为

$$\left.\begin{aligned}\frac{\partial\sigma_x}{\partial x}+\frac{\partial\tau_{xy}}{\partial y}+f_x&=0\\\frac{\partial\tau_{xy}}{\partial x}+\frac{\partial\sigma_y}{\partial y}+f_y&=0\end{aligned}\right\}\quad(x,y)\in\Omega_1\cup\Omega_2\tag{1}$$

外力边界条件为

$$\left.\begin{aligned}\sigma_x\cos(n,x)+\tau_{xy}\cos(n,y)&=\varphi_x\\\tau_{xy}\cos(n,x)+\sigma_y\cos(n,y)&=\varphi_y\end{aligned}\right\}\quad(x,y)\in\Gamma_1\tag{2}$$

位移边界条件为

$$\left.\begin{aligned}u&=\bar{u}\\v&=\bar{v}\end{aligned}\right\}\quad(x,y)\in\Gamma_2\tag{3}$$

弹性系数间断边界条件为

$$\left.\begin{array}{l}(\sigma_x\cos(n,x)+\tau_{xy}\cos(n,y))^+ \\ \quad =(\sigma_x\cos(n,x)+\tau_{xy}\cos(n,y))^- \\ (\tau_{xy}\cos(n,x)+\sigma_y\cos(n,y))^+ \\ \quad =(\tau_{xy}\cos(n,x)+\sigma_y\cos(n,y))^-\end{array}\right\}(x,y)\in\Gamma_0 \tag{4}$$

上述偏微分方程 (1) 及边界条件 (2) ~ (4) 等价于变分原理 (**最小势能原理**)

$$J(u,v)=\frac{1}{2}\iint_{\Omega_1\cup\Omega_2}\boldsymbol{\sigma}^{\mathrm{T}}\boldsymbol{\varepsilon}\mathrm{d}x\mathrm{d}y-\iint_{\Omega_1\cup\Omega_2}\boldsymbol{f}^{\mathrm{T}}\boldsymbol{\delta}\mathrm{d}x\mathrm{d}y-\int_{\Gamma_2}\boldsymbol{\varphi}^{\mathrm{T}}\boldsymbol{\delta}\mathrm{d}s=\min \tag{5}$$

这里

$$\boldsymbol{\sigma}=\begin{Bmatrix}\sigma_x\\ \sigma_y\\ \tau_{xy}\end{Bmatrix}=\boldsymbol{D}\begin{Bmatrix}\varepsilon_x\\ \varepsilon_y\\ \gamma_{xy}\end{Bmatrix},\quad \boldsymbol{\varepsilon}=\begin{Bmatrix}\varepsilon_x\\ \varepsilon_y\\ \gamma_{xy}\end{Bmatrix}=\begin{Bmatrix}\dfrac{\partial u}{\partial x}\\ \dfrac{\partial v}{\partial y}\\ \dfrac{\partial u}{\partial y}+\dfrac{\partial v}{\partial x}\end{Bmatrix} \tag{6}$$

分别为应力应变分量,$\boldsymbol{\delta}=\begin{Bmatrix}u\\ v\end{Bmatrix}$为位移分量,$\boldsymbol{D}$ 是根据虎克定律得到的弹性矩阵,$\boldsymbol{f}=\begin{Bmatrix}f_x\\ f_y\end{Bmatrix}$,$\boldsymbol{\varphi}=\begin{Bmatrix}\varphi_x\\ \varphi_y\end{Bmatrix}$分别为给定的体积力和 Γ_1 上的边界力,($\cos(n,x)$, $\cos(n,y)$) 为边界曲线法线的方向余弦,弹性系数在 Ω_1,Ω_2 内可分别有不同的值,对式 (5) 中的 u,v 取极小,则满足式 (5) 的 u,v 除了在区域 $\Omega_1\cup\Omega_2$ 内满足平衡方程 (1) 外,在边界 Γ_1,Γ_0 自动满足边界条件 (2) ~ (4)。

剖分与插值 为适于电子计算机数值求解,还必须对上述式 (5) 进行离散化,为此,需对求解区域 Ω 进行单元剖分,在每个单元内用一个较简单的插值函数代替未知函数 u,v,在二维区域内最基本最简单的单元是三角形单元 (见图 1)。在曲线边界上则裁弯取直,用折线代替曲线,当然还可以取四边形单元或曲边的三角形和四边形单元等,这些单元的顶点可称为网格结点。插值函数依据不同的单元类型有不同的取法,对于简单的三角形单元,使用线性插值函数,即在每个三角形单元 Δ_k 内,用 $U(x,y)=a_kx+b_ky+c_k$ 来近似代替 $u(x,y)$,其中系数 (a_k, b_k, c_k) 由条件 $U_i=U(x_i,y_i)=U_i(i=1,2,3)$ 来确定。这样在区域 Ω 内部,未知函数 $u(x,y)$ 用一个分片线性函数 $U(x,y)$ 代替,对于未知函数 $v(x,y)$ 按同样方法处理。

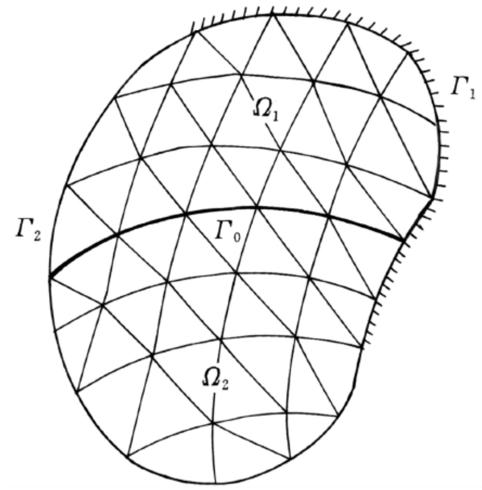


图 1 三角形单元

有限元的离散化 上述插值函数 $U(x,y)$, $V(x,y)$ 取定以后,将其代入式(6),则有

$$\boldsymbol{\varepsilon}=\begin{Bmatrix}\varepsilon_x\\ \varepsilon_y\\ \gamma_{xy}\end{Bmatrix}=\boldsymbol{B\Delta},\quad \boldsymbol{\sigma}=\boldsymbol{D\varepsilon}=\boldsymbol{DB\Delta} \tag{7}$$

这里 $\boldsymbol{B}$ 为应变–位移矩阵,$\boldsymbol{\Delta}$ 为所有网格结点上位移值 $\begin{Bmatrix}U_i\\ V_i\end{Bmatrix}$ 组成的向量,把式 (7) 代入式 (5) 有

$$\begin{aligned}J(u,v)\approx J(U,V)&=\frac{1}{2}\iint_{\Omega}\boldsymbol{\varepsilon}^{\mathrm{T}}\boldsymbol{D\varepsilon}\mathrm{d}x\mathrm{d}y-\boldsymbol{\Delta}^{\mathrm{T}}\boldsymbol{F}\\ &=\frac{1}{2}\boldsymbol{\Delta}^{\mathrm{T}}\boldsymbol{K\Delta}-\boldsymbol{\Delta}^{\mathrm{T}}\boldsymbol{F}\end{aligned} \tag{8}$$

这里

$$\boldsymbol{K}=\iint_{\Omega}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{DB}\mathrm{d}x\mathrm{d}y=\sum_e\iint_{\Omega_e}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{DB}\mathrm{d}x\mathrm{d}y=\sum_e\boldsymbol{K}_e$$

式中,$\boldsymbol{K}_e$ 称为单元刚度矩阵,$\boldsymbol{K}$ 为整体刚度矩阵,它是由各单元刚度矩阵叠加而得,$\boldsymbol{F}$ 是由式 (5) 中第二、三项离散后得到的外力向量,在式 (8) 中对 $\boldsymbol{\Delta}$ 中所有分量求偏导数,并令其等于 0,得到下面的线性代数方程组

$$\boldsymbol{K\Delta}=\boldsymbol{F} \tag{9}$$

从上面过程可以看到,在变分原理及剖分插值函数取定以后,线性方程组 (9) 即完全确定,$\boldsymbol{K}$ 和 $\boldsymbol{F}$ 的具体计算由电子计算机以统一的方式完成,可以很方便地进行计算机程序编制。

有限元数值解法的应用 从上面得到的线性代数方程组 (9),其系数矩阵 $\boldsymbol{K}$ 有许多好的性质,如对称、正定和高度稀疏性,这就为数值求解提供了很好的条件,求解这种线性方程组的方法,归纳起来有

两类：一类是直接解法，如线性方程组的高斯消去法、平方根法。另一类是迭代解法，如逐次超松弛法、切比雪夫半迭代以及带预条件 z 的共轭斜量法（PCG 法）均是目前常用的有效的数值解法。当求得结点位移以后，将其代入应力-应变-位移关系式（7），可得到工程设计中最感兴趣的各个单元应力和结点应力值。

由于有限元方法能方便求解各种比较复杂的问题，加之计算机的飞速发展，大容量的存储及快速运算速度，促进了有限元方法在几乎所有工程部门（如航空航天、水工建筑、机械电子、石油化工以及国防工业部门）的应用，而针对各个工程部门研制出了各种大型通用和专用有限元软件包（如美国的 ADINA，ANSYS，NASTRAN，德国的 ASKA 等），与现代先进的计算机辅助设计技术（CAD）相结合，大大地缩短了工程设计周期，极大地推动了生产力发展与科学技术进步。

有限元方法最早由 R. Courant 于 1943 年在一篇论文中提出，由于其运算量大，没有受到重视和得到应用发展。直到 20 世纪 50 年代中期，随着电子计算机出现，在西方首先由工程师们应用于航空结构分析，以后随着计算机技术的发展而逐步应用于其他工程领域。60 年代后期，数学家们也相继参加到有限元方法研究中来，逐步建立了有限元方法严格的数学理论。在这里应当提到的是我国科技工作者对有限元方法应用与发展也作出了重要贡献。60 年代初，我国冯康教授及其领导下的科研组在解决大型水坝计算中，独立于西方发展了有限元方法，并且先于西方国家建立了严格的有限元的数学理论基础。以后我国许多科技工作者又在许多具体方向上，作出了很多有价值的贡献。

参考文献

1. Zienkiewicz O Z. The finite element method. 3rd ed. London: McGraw-Hill, 1977

2. 冯康. 数值计算方法. 北京：国防工业出版社，1978

（王荩贤）

youxian zidongji

有限自动机（finite automaton，FA）　有限离散数字系统的抽象数学模型。加法器、奇偶校验器、模 3 计数器、二单位时间延迟器、自动电话交换机、升降机、计算机操作系统进程的活动过程、计算机、时序电路、开关网络等都是有限离散数字系统的实例。主要研究系统的综合与分析，即给出具体的功能要求，综合建成能实现这些功能的有限自动机并设计出满足要求的逻辑网络，给出有限自动机或其实现网络，分析有限自动机的功能描述，时间复杂度，故障传播与检测。

有限自动机的类型与等价　确定型有限自动机（DFA）的数学定义为五元组 $M=(Q,\Sigma,\delta,q_0,F)$，其中 Q 是状态的有限集，Σ 是输入字母的有限集，$q_0\in Q$ 称为初（始状）态，$F\subseteq Q$ 为终（止状）态集，转移函数 $\delta:Q\times\Sigma\to Q$ 是单值函数，即对任一输入来说，在给定状态下有唯一的下一步的状态。实际问题中常常要求处理字母的有限序列（包括空序列 ε，或称空字），故 δ 的定义范围往往扩大到集合 Σ^* 上。若 $\delta:Q\times\Sigma^*\to Q$ 是（全）函数，称 M 为完全的确定型有限自动机；若 δ 是部分（偏）函数（对有些状态无定义）称为不完全的（偏）确定型有限自动机。若 $\delta:Q\times\Sigma^*\to 2^Q$（$2^Q$ 表示 Q 的所有子集的集合即幂集），即 δ 为多值函数，称 M 为非确定有限自动机（NFA），也可分为完全的和偏的两种。若 $\delta:Q\times(\Sigma\cup\{\varepsilon\})\to 2^Q$，称 M 为带 ε-转移的 NFA。若 $\delta:Q\times\Sigma\to Q\times\{L,R\}$ 称 M 为双向有限自动机，其中 L,R 分别表示扫描输入字母时读头左、右移一个位置（2DFA）。已经证明 DFA、NFA、带 ε-转移的 NFA、2DFA 等价，即功能相同。实际问题（如寄存器、计数器、信号转换器等）常常需要考虑输出的结果和状态，故研究了时序机械装置——两类带输出的有限自动机。①六元组 $M=(Q,\Sigma,\Delta,\delta,\lambda,q_0)$ 称为摩尔机（状态赋值机），其中 Q,Σ,δ,q_0 与 DFA 中相同，Δ 为输出字母的有限集，$\lambda:Q\to\Delta$ 是单值函数，即输出只与到达的状态有关，与从哪个状态转移而来无关。输入 ε 时，输出为 $\lambda(q_0)$，输入 $a_1a_2\cdots a_n$ 时输出为 $\lambda(q_0)\lambda(q_1)\cdots\lambda(q_n)$，其中 $\delta(q_{i-1},a_i)=q_i$，$1\leqslant i\leqslant n$。DFA 可以看作 $\Delta=\{$接收，拒绝$\}$时的摩尔机。②六元组 $M=(Q,\Sigma,\Delta,\delta,\lambda,q_0)$，称为米林机（转换赋值机），其中 $\lambda:Q\times\Sigma\to\Delta$，其余与摩尔机中相同。输入 ε 时输出也为 ε，输入 $a_1a_2\cdots a_n$ 时，输出为 $\lambda(q_0,a_1)\lambda(q_1,a_2)\cdots\lambda(q_{n-1},a_n)$，其中 $q_i=\delta(q_{i-1},a_i)$，$1\leqslant i\leqslant n$，已知摩尔机与米林机等价。

有限自动机有两种表示法，一是状态表或转移矩阵，一是有向图（称为状态图用⓪表示终态）。例如奇偶校验器可设计为一台米林机 $M_1=(\{A,B\},\{0,1\},\{0,1\},\delta,\lambda,A)$，输入串有奇数个 1 时输出 1，输入串有偶数个 1 时（此时状态为 A）输出 0。M_1 的状态表和状态图如图 1 所示。

米林机状态图中弧上带的符号为输入输出。

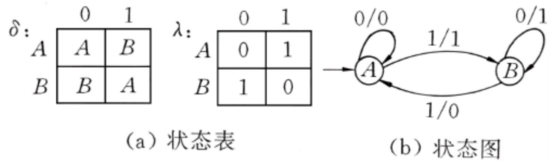


图 1　M_1 的状态表和状态图

从上述米林机可设计一台摩尔机 $M_2=(\{(A,0),(B,1)\},\{0,1\},\{0,1\},\delta,\lambda,(A,0))$。$M_2$ 的状态表和状态图如图 2 所示。

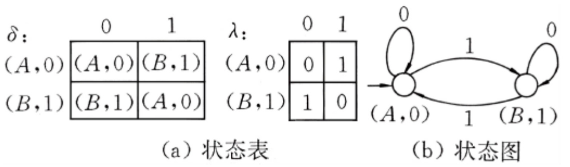


图 2　M_2 的状态表和状态图

状态化简　实际问题迫切需要设计出功能相同而状态最少的系统,这就是有限自动机的极小问题。对完全确定有限自动机的状态集引入关系“≡”,对每一输入串 x,若 $\delta(p,x)\in F$ 当且仅当 $\delta(q,x)\in F$,则称状态 $p\equiv q$,“≡”是一个等价关系,从而可将 Q 划分为若干个等价类,凡在同一个等价类的状态具有相同的功能,不在同一等价类的状态功能不同,于是可简化地视一个等价类为一个状态,从而完全的确定型有限自动机在同构的意义下有唯一的与它功能相同而状态数最少的有限自动机存在,从而给出了极小化的有效算法。

时序网络与有限自动机的模拟　组成时序网络的基本逻辑元件有组合元件与记忆元件两类。组合元件在时刻 t 的输入完全决定了时刻 t 的输出,如与、或、非、与非、或非等阈电路元件。记忆元件在时刻 t 的输出与时刻 t 的输入与状态有关,如各种触发器和延迟器。去掉时序网络中连接记忆元件的网络输入线后不再含有回路,称之为合式网络,无记忆元件的合式网络称为组合网络。用布尔代数或卡诺图可简化描述输入输出方程组,设计出符合要求、结构简单的二值元件组合网络。合式网络需要用极小化有限自动机。具有同样输入集和输出集的两个有限自动机 M_1 和 M_2 的状态 q_1 和 q_2,若对所有输入序列都有相同的输出,称 q_1 与 q_2 等价。如果 $M_1(M_2)$ 的每个状态总有 $M_2(M_1)$ 的一个状态与之等价,称 M_1 与 M_2 等价,即 M_1 与 M_2 功能相同。现作进一步推广,若有穷自动机 $M_1=(Q_1,\Sigma_1,\Delta_1,\delta_1,\lambda_1,q_0)$ 与 $M_2=(Q_2,\Sigma_2,\Delta_2,\delta_2,\lambda_2,q_0')$ 之间存在映射 $h_1:Q_2\to Q_1$,映射 $h_2:\Sigma_2\to\Sigma_1$,$h_3:\ \Delta_1\to\Delta_2$,且对 $a_2\in\Sigma_2$,$q_2\in Q_2$ 时有

$$\begin{aligned}h_1^{-1}(\delta_1(h_1(q_2),h_2(a_2)))&=\delta_2(q_2,a_2)\\ h_3(\lambda_1(h_1(q_2),h_2(a_2)))&=\lambda_2(q_2,a_2)\end{aligned}\qquad(1)$$

称 M_1 是 M_2 的一个模拟。映射 h_2 是一个编码器,它将 M_2 的输入信息变成 M_1 的输入信息,然后再输入 M_1,应用转移函数 δ_1 处理后输出,经译码器 h_3 译成 M_2 的输出信息。若满足式(1),M_1 就是 M_2 的一个模拟。h_2,h_3 不同,模拟就有所不同。反映在有限自动机模拟时序电路也就不同,即模拟时对每一状态指定一组代表基本元件状态组合的编码,将编码分配给各状态。不同的分配方案产生不同的逻辑网络,各有其复杂程度。选择好的状态分配方案使逻辑网络的构造尽可能简单,是模拟的主要研究课题之一。加上延迟器的异步网络,考虑基本元件状态组合的编码分配时,还需要注意状态转换的稳定性,保证无矛盾、无竞争。

有限接收器　不考虑输出的有限自动机 $M=(Q,\Sigma,\delta,I,F)$,其中 I 为初态集,字 $x\in\Sigma^*$ 合于 $\delta(q_0,x)\in F$,$q_0\in I$,称 x 为 M 所接收或 M 能识别 x。$L(M)=\{x\in\Sigma^*\mid\delta(q_0,x)\in F,q_0\in I\}$ 称为 M 所接收(识别)的语言,是正规语言(它由某一**正规文法**产生,与正规集或正规表达式等价)。任一正规语言可以由一个带 ε-转移的不确定有限自动机所接收。有限自动机和有限接收器有三点不同:①前者只有一个初态;后者为非空初态集。②前者主要考虑对输入的响应;后者考虑的是输入后的状态是否属于终态集,以便判断该输入是否被接收。③前者的转移函数往往是单值的;后者是多值的且可能是偏函数。

在与形式语言**乔姆斯基分层**对应的自动机分层中,有限自动机类是**下推自动机**类的真子集。有限自动机经布尔运算(并、交、补)、替换、同态、逆同态、商、MAX、MIN、CYCLE、INIT、reversal、1/2、SQRT、LOG 等运算后仍为有限自动机。

神经网络　历史上最早用有限自动机模型化的系统。每一个神经元由细胞体、神经纤维和突触组成,细胞体经神经纤维发出若干突触。有限个神经元连接在一起构成神经网络。连接时,每个神经元的任一突触只接触一个细胞体。每一突触只有兴奋(用圈表示)和抑制(用点表示)两种状态。若输入 1,兴奋突触个数超过抑制突触的个数,至少为该细胞体(用三角形表示)的阈值(置于细胞体中),神经元就输出 1。假定有足够的时间改变输入值使信号

传播、网络达到稳定的配置,则等价于图 3 所示神经网络行为的有限自动机(y_1,y_2,y_3 的初值设为 0)如图 4 所示。

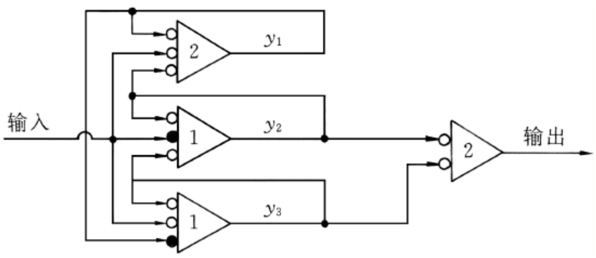
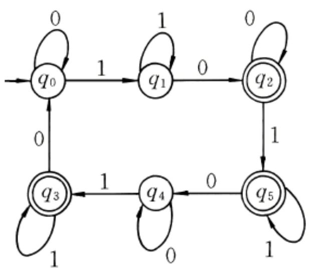


图 3　神经网络

	y_1	y_2	y_3	0	1
q_0	0	0	0	q_0	q_1
q_1	0	0	1	q_2	q_1
q_2	0	1	1	q_2	q_5
q_3	1	0	0	q_0	q_3
q_4	1	1	0	q_4	q_3
q_5	1	1	1	q_4	q_5

(a) 状态表



(b) 状态图

图 4　与图 3 相应的有限自动机

有限自动机的应用　计算机程序语言编译中,话前缀的集合是正规语言,可以被生成该语言之文法的项目作为状态并另加一初态的非确定的有限自动机所接收,然后找出等价的极小确定有限自动机,从而节约词法分析的时间和空间。又在编译程序里可完成扫描器的任务,识别变元名字、运算符、常数、标识符等。例如长度限制为 6 的 FORTRAN 语言标识符和 \$ 以及大写英文字母可表示为正则表达式

$$(\$ + A + B + \cdots + Z)(\varepsilon + \$ + A + B + \cdots + Z + 0 + 1 + \cdots + 9)^5$$

转移函数用二维数组编码表表示,终态表明已找到特殊的符号。

文本编辑程序也与词法分析的情形类似,将字串与一正规表达式相匹配,求接收它的带 ε-转移的非确定有限自动机,一旦列举该自动机所有状态的列构造好,就使之在输入的前缀上进入,前面的列不再需要便舍弃掉以节约空间。可应用于图书资料索引查找、结构型数据翻译等。

是否能够由输出决定相应的输入呢?这个问题导致研究有限自动机的可逆性。现已得到可逆的充分必要条件是有限自动机的状态图 G 没有回路,G 的层次为 ρ,则自动机延迟 $\rho+1$ 步可逆,且对任何 $\tau\leq\rho$,不能延迟 τ 步可逆。

当有限自动机 M 的输入、输出、状态集分别为有限域上 l,m,n 维向量空间时,称 M 为线性有限自动机。当有限自动机 $M=(Q,\Sigma,\Delta,\delta,\lambda,q_0)$ 的转移函数 δ 和输出函数 λ 不依赖于输入时,称为自治有限自动机。对它们也证明了存在其极小化。对编码、移位寄存器、通信的可靠性、保密性的研究有其作用。近年来也研究了状态识别试验,即在不知道有限自动机 M 的内部状态时,通过馈入序列来观察分析 M 的结构与功能,现已发展成暗箱理论。

也有人去掉输入字长有限这一条件,研究 **ω-有限自动机**,平行地得到了一些结果。

参考文献

Hopcroft J E, Ullman J D. Introduction to automata theory, languages, and computation. Reading, MA: Addison-Wesley, 1979

(张一立)

youxiang tu

有向图(directed graph)　每条边均为有向边的图(参见**图论**)。设有向图 $G=\langle V,E,\psi\rangle$,对任意顶点 $v\in V$,称以 v 为终点的边的数目为 v 的**入度**,记为 $d_G^-(v)$。称以 v 为起点的边的数目为 v 的**出度**,记为 $d_G^+(v)$。称 $d_G^+(v)+d_G^-(v)$ 为 v 的度,记为 $d_G(v)$。如果 G 中任意两顶点都相互可达,则称 G 是强连通的。G 的极大强连通子图称为 G 的强分支。如果对于 G 的任意两顶点,必有一个顶点可达另一个顶点,则称 G 是单向连通的。G 的极大单向连通子图称为 G 的单向分支。如果 G 的底图是连通的,则称 G 是**弱连通**的。G 的极大弱连通子图称为 G 的弱分支。设 $n\in I_+$(正整数集合),每个顶点的出度和入度均为 $n-1$ 的 n 阶简单有向图称为完全有向图。设 $V=\{v_1,v_2,\cdots,v_n\}$ 且 $E=\{e_1,e_2,\cdots,e_m\}$,G 的邻接矩阵 $\boldsymbol{X}(G)$ 是一个 $n\times n$ 矩阵 (x_{ij}),其中 x_{ij} 为以 v_i 为起点和以 v_j 为终点的边的数目。G 的路径矩阵(或称可达性矩阵)$\boldsymbol{P}(G)$ 是一个 $n\times n$ 矩阵 (p_{ij}),其中,

$$p_{ij}=\begin{cases}1, & \text{若从 } v_i \text{ 可达 } v_j\\ 0, & \text{否则}\end{cases}$$

若 G 无自圈,则 G 的关联矩阵 $\boldsymbol{A}(G)$ 是一个 $n\times m$ 矩阵 (a_{ij}),其中

$$a_{ij}=\begin{cases}1, & \text{若 } v_i \text{ 是 } e_j \text{ 的起点}\\ -1, & \text{若 } v_i \text{ 是 } e_j \text{ 的终点}\\ 0, & \text{否则}\end{cases}$$

当然，G 还有其他的矩阵表示。矩阵表示很适合用计算机表示有向图和对有向图进行操作。 （张强）

youxu eryuan juecetu

有序二元决策图（ordered binary decision diagrams，OBDD） 表示布尔函数的数据结构的一种有向无环图。图中变量被指定以一定的先后次序，可以实现对布尔函数的有效表示和操作运算。有序二元决策图亦可用于表示集合、关系、图等数据结构。

1978 年，Sheldon B. Akers 正式提出二元决策图（BDD），将布尔函数表示为有根的有向无环图。图中结点分为非终止结点和终止结点：终止结点分为 0-终止结点和 1-终止结点，分别表示常量 0 和 1；任意非终止结点 u 都被标记为一个布尔变量（记为 $var(u)$），包含两个子结点，称为 low 孩子和 $high$ 孩子，记为 $low(u)$ 和 $high(u)$，连接到子结点的边（用虚线和实线表示）分别表示将 $var(u)$ 赋值为 0 和 1。因此，一条从根结点到 0-终止结点（1-终止结点）的路径就表示使得 BDD 对应的布尔函数表达式的值为 0(1) 的一组变量赋值。在此基础上，1986 年，Randal E. Bryant 通过对 BDD 中任意路径上出现的变量附加一个线性序提出了 OBDD。设 X 是布尔函数中出现的变量集合，$<$ 是 X 上的一个线性序，若 BDD 中任意两个非终止结点 u 和 v 都满足 $var(u) < var(v)$，则称该 BDD 为有序二元决策图。图 1 表示的是一个 OBDD，其中结点变量顺序为 $x_1 < x_2 < x_3 < x_4$。

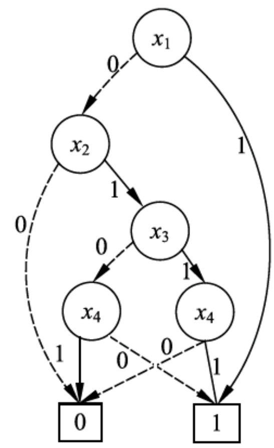


图 1　有序二元决策图的例子

一个 OBDD 中，可能存在如下两种冗余情况：

（1）两个非终止结点 u 与 v 同构，即 $var(u) = var(v)$ 且 $low(u) = low(v)$ 且 $high(u) = high(v)$，如图 2(a) 所示。

（2）非终止结点 u 指向相同的子结点，即 $low(u) = high(u)$，如图 2(b) 所示。

去掉上述冗余情况的 OBDD 称为精简有序二元决策图（reduced OBDD，ROBDD），有时也称 OBDD。任意给定 OBDD 中的所有冗余结点都能在线性时间内删除。

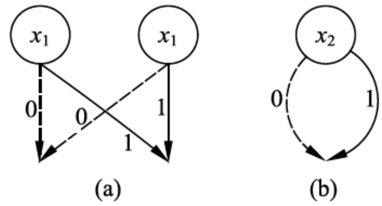


图 2　OBDD 中的结点冗余的情况

OBDD 主要具有如下优点：①在给定变量次序下，任意布尔函数对应的 ROBDD 唯一，且其规模在所有等价的 OBDD 中最小；②能在多项式时间内支持多种布尔运算，如否定、合取、析取等。OBDD 表示常用于模型检测、智能规划、软件和硬件的形式化验证、本体推理、**数据挖掘**等计算机科学研究领域。

目前，OBDD 领域存在的主要问题是对于多数布尔函数，其对应的 ROBDD 的规模为指数级的，这导致了对于某些实际问题，使用 OBDD 表示的空间代价不可承受。解决思路是对 OBDD 中条件进行适当放松，提出了多种替代方案，例如，FBDD、DNNF、d-DNNF 等，这些方法的优点是能更紧致地表示布尔函数，其缺点是对于一些布尔运算不再存在多项式时间的算法，因此，如何平衡空间有效性和易处理性仍然是今后研究的重要课题。

参考文献

1. Bryant R E. Graph-based algorithms for Boolean function manipulation. IEEE Transactions on Computers, 1986, 8(C-35): 677-691

2. Bryant R E. Symbolic Boolean manipulation with ordered binary decision diagrams. ACM Computing Surveys, 1992, 24(3): 293-318

（欧阳丹彤）

yufa

语法（syntax） 构建语言的结构正确成分所需遵循的规则的集合。**程序设计语言**的语法定义阐明了何种字符串构成该语言的一个合法的程序。早期语言的语法是用自然语言描述的，如早期的 **FORTRAN 语言**。从 **ALGOL 60 语言**开始，人们都用形

式方法描述语法。描述语法的常用形式描述工具是**BNF**,扩充的**BNF**和**语法图**。　(程虎)

yufa fenxi

语法分析(syntax analysis,parsing)　按语言的语法规则分析和处理由词法分析程序产生的词牌流并转换成语法分析树的过程。

程序设计语言的语法一般是上下文无关文法,即 chomsky 2 型文法。通常采用 BNF 或**语法图**来描述。

语法分析是编译过程的一个阶段,其输入为词牌流,输出为语法分析树的某种表示及有关信息。有些编译程序,在语法分析过程中除对源程序进行语法检查外,还进行静态语义检查。当检查中发现不符合规则的情况时,系统应指明错误的性质和可能的错误位置,以供用户改错时参考。

实现语法分析的算法有多种,如算符优先法、递归下降法和 LR 法等。其中一类为自顶向下分析方法,即从树根(程序)向树叶(即终结符)方向进行分析。递归下降法是一种典型的自顶向下分析方法。另一类为自底向上分析方法,即从树叶向树根方向进行归约。算符优先法是一种典型的用来处理表达式的自底向上分析方法。LR 分析法可以认为是自左至右、自底向上的移进-归约分析的高度概括和集中,它比算符优先法或其他"移进-归约"技术更加广泛,而且识别效率并不比它们差。

参考文献

1. Aho A V, Sethi R, Ullman J D. Compilers principles, techniques and tools. Addison-Wesley, 1986

2. 陈火旺,钱家骅,孙永强. 程序设计语言编译原理. 2 版. 北京:国防工业出版社,1984

(钱树人)

yufatu

语法图(syntax diagram)　语法的图形描述。每个图描述一种非终极符号的定义。图中用圆圈表示终极符号,用方框表示非终极符号,用有向弧表示走向,图上一条通路就表示该语法结构的一种正确定义方式。

例如,某语言的标识符可描述为图 1(a),表示由字母开头的字母数字串组成,字母和数字最多总共可有 6 个。字母可描述为图 1(b),数字可描述为图 1(c)。　(程虎)

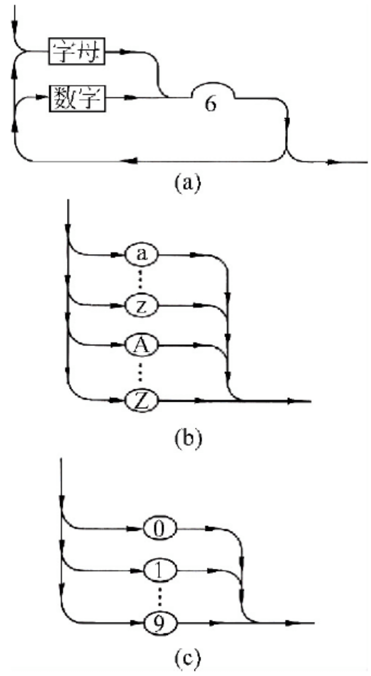


图 1　语法图

yuju

语句(statement)　**高级语言**中用于表达处理动作的基本单位,用以描述程序中的运算步骤、控制构造及数据传输。

从语句的构造方式来看,语句可分为简单语句与构造语句。简单语句包括空语句,赋值语句,过程调用语句(又称过程语句),转向语句,出口语句,返回语句等;构造语句包括复合语句,重复语句,条件语句等。

赋值语句将表达式的值作为左部变量的当前值。其形式是

变量名: =表达式　(PASCAL 语言形式)

变量名=表达式　(FORTRAN 语言形式)

一般说来,要求赋值变量的类型与右部表达式值的类型相同或赋值相容。

过程语句表示对过程名所标记的过程的一次调用,其形式是

过程名(实在参数表)

实在参数表可以无参数,也可以有一个或多个参数。

过程语句的执行首先检查参数表是否一致,即其实在参数表中的实参与过程说明中的形参在个数、顺序、类型上是否保持一致。若一致,则传递参数,然后去激活相应的过程体(参见**函数与过程**)。

空语句表示空运算;转向语句为用以改变程序

执行次序的语句。

构造语句是由简单语句按一定规则构造起来的语句。

复合语句是将若干个语句用 begin 和 end 括起来的一个语句。程序执行时按书写的顺序执行复合语句内的各个成分语句。

条件语句一般又可分为如果语句与情况语句两种。如果语句主要用于处理有两个分支的选择结构,而情况语句主要用于多分支选择结构。

如果语句有 if-then 和 if-then-else 两种形式。

如果语句规定:在一定条件下执行某一种成分语句,否则执行另一种成分语句;或者不执行任何成分语句。

情况语句的一般形式为:

```
case   表达式   of
    情况常量表 1:语句 1;
    情况常量表 2:语句 2;
      …
    情况常量表 n:语句 n
end
```

情况语句就是根据情况表达式的取值,选择其值与情况常量表的某一情况常量相同的成分语句执行,该成分语句执行后,便去执行该情况语句后的语句。

重复语句规定一些语句被重复地执行。被重复的语句称为重复体。重复语句包括 while 语句, Repeat 语句和 for 语句。

while 语句的一般形式为:

```
  while 条件 do 语句
```

其中,条件为一布尔表达式,do 后面的语句是 while 语句的重复体,它可以是简单语句,也可以是复合语句。

Repeat 语句的一般形式为:

```
Repeat
  语句 1;
  语句 2;        重复体
  …
  语句 n
until 条件   (布尔表达式)
```

重复执行重复体内的语句,直到 until 后的布尔表达式的条件成立为止。

for 语句有 for-to-do 和 for-down to-do 两种形式。

for 语句是预先给出重复次数的一种重复语句,其重复次数由指定的初值和终值决定,并由循环控制变量来控制。循环控制变量在指定的值域[初值,终值]内顺序取值(分递增与递减两种方式),每取一次值就执行一次重复体,直至该值域内的值全被访问,重复结束。

从语句的执行方式来看,又可分为顺序语句与并发语句两类,与顺序执行的语句不同,并发语句用来描述若干个并发的计算活动,这些活动的总体效果与活动开始的次序或活动执行的次序无关。引入并发语句的第一个语言为 P. B. Hansen 开发的并发 Pascal,它是 **Pascal 语言**的一个变种。语言中的复合语句,子程序及程序按顺序语言的分程序结构方式组织起来,但处理部件要等确立并发性的其他程序触发之后,并让它们真正启动才可以执行。以下是一个由三个过程组成的复合体的例子,它们中一个或全部均能并行地执行。

```
type P1 = Process (相关部件表)
Var       P1 的说明
begin     P1 的活动       P1 定义
end.
type P2 = Process (相关部件表)
Var       P2 的说明
begin     P2 的活动       P2 定义
end.
type P3 = Process (相关部件表)
Var       P3 的说明
begin     P3 的活动
end.
Var       Pname1:P1;        Pname2:P2;
          Pname3:P3;        激活并发进程
begin init Pname1,Pname2,   Pname1,Pname2,
          Pname3;           Pname3.
end.
```

这里 Pname1,Pname2 及 Pname3 分别为 P1 类型,P2 类型及 P3 类型的实例。

早期有些高级语言(如 FORTRAN Ⅳ)中,把语言中不是用来表达算法,而是用来描述数据对象的定义的说明性成分也看作为语句,把这类语句称为不可执行语句,并把以前的表达动作行为的语句称为可执行语句。现代高级语言中,一般将这类说明性的不可执行语句列入语言的说明部分(参见**说明**)。

参考文献

Ralston A, Reilly E D. Encyclopedia of computer science. 3rd ed. New York: Van Nostrand Reinhold, 1992

(陈涵生)

yuliaoku yuyanxue

语料库语言学（corpus linguistics）　研究自然语言机读文本（或称“电子文本”）的采集、存储、标注、检索、统计等方法的一门学科。其目的是通过对客观存在的大规模真实文本中的语言事实进行定量分析，为语言学研究或自然语言处理系统开发提供支持。应用领域包括：语言文字的计量分析、语言知识获取、作品风格分析、词典编纂、全文检索和**机器翻译**等各种自然语言处理系统。

语料库语言学的起源可以追溯到20世纪50年代美国Leonard Bloomfield后期的结构主义语言学时代，那时的语言学家在科学的实证主义和行为主义观点的影响下，认为语料库是一个规模足够大的语言数据库。这些语言数据是自然发生的，对于语言学研究是必要和充分的，而直觉证据充其量只是一种贫乏的第二位的资源。50年代后期Noam Chomsky创建了转换生成语法，他主张直觉是合理的，而任何自然的语料都是扭曲的。他的这种理性主义的观点构成了当代理论语言学家的正统观念，在极大程度上扼制了早期语料库语言学的发展。语料库语言学在20世纪80年代后期再度崛起的主要原因是：①基于规则的句法-语义分析方法赖以利用的语言知识无论是词典信息还是语法规则，主要通过语言学家的内省来获取，而实际上这种知识不可能覆盖真实文本中出现的所有语言事实；②计算机和计算技术的迅猛发展，使语料库的规模急速增长，从早期的百万词次猛增到数亿词次。这是以往语言学家都未曾预料到的。这一事实使得语言的词汇、句法等任何现象都能够凭借语料库来进行开放性的调查。

20世纪60年代初建立的现代美国英语的布朗语料库和70年代创建的现代英国英语的LOB语料库被称为第一代语料库，库容量都是100万词次。80年代，有了光学字符识别（OCR）技术，语料库规模迅速增长，规模都在千万词次，如COBUILD语料库，2000万词次；Longman/Lancaster英语语料库，3000万词次。这一时期建立的语料库被称为第二代语料库。进入90年代后，由于词处理编辑软件和桌面印刷系统的普及，尤其是**互联网**技术的迅速发展和普及，数量巨大的电子文本成为语料库取之不竭的资源，随之出现的第三代语料库规模达到数亿词次，如英国的牛津文本档案库等。

语料库语言学研究的主要内容包括：①基本语料库的建设；②语料加工工具和标注方法的研究；③语料库管理；④从语料库中获取语言知识的技术与方法。

语料库已经成为自然语言处理研究和应用系统开发不可或缺的基础，自然语言处理技术的发展反过来又进一步加速了语料库语言学的发展。

参考文献

1. Aijmer K，Altenberg B. English corpus linguistics. London：Longman，1991

2. 黄昌宁，李涓子. 语料库语言学. 北京：商务印书馆，2002

3. 梁茂成，李文中，许家金. 语料库应用教程. 北京：外语教学与研究出版社，2010

（宗成庆　黄昌宁）

yuyan chuli xitong

语言处理系统（language processing system）　对**软件语言**进行处理的程序系统。

除了**机器语言**外，其他用任何软件语言书写的程序都不能直接在计算机上执行，都需要对它们进行适当的处理。语言处理系统的作用是把用软件语言书写的各种程序处理成可在计算机上执行的程序，或最终的计算结果，或其他中间形式。

不同级别的软件语言有不同的处理方法和处理过程。关于需求级、功能级、设计级和文档级软件语言的处理方法和处理过程是软件语言、**软件工具**和**软件开发环境**的重要研究内容之一。关于实现级语言即**程序设计语言**的处理方法和处理过程发展较早，技术较为成熟，其处理系统是基本软件系统之一。这里，语言处理系统仅针对程序设计语言的处理而言。关于需求级、功能级、设计级和文档级语言的处理请参见**需求定义语言**，**功能性语言**，**设计性语言**，**软件过程**和**软件工具**。

程序设计语言处理系统随被处理的语言及其处理方法和处理过程的不同而异。不过，任何一个语言处理系统通常都包含有一个**翻译程序**，它把一种语言的程序翻译成等价的另一种语言的程序。被翻译的语言和程序分别称为源语言和源程序，翻译生成的语言和程序分别称为目标语言和目标程序。按照不同的源语言、目标语言和翻译处理方法，可把翻译程序分成若干种类。从**汇编语言**到机器语言的翻译程序称为**汇编程序**，从**高级语言**到机器语言或汇编语言的翻译程序称为**编译程序**。按源程序中指令

或语句的动态执行顺序,逐条翻译并立即解释执行相应功能的处理程序称为**解释程序**。除了翻译程序外,语言处理系统通常还包括正文编辑程序、**宏加工程序**,**连接编辑程序**和**装入程序**等。

发展过程

随着程序设计语言的变化和发展,语言处理系统也跟着由小到大、由简单到复杂的变化和发展。最初人们直接用机器语言来描述问题的解法,这种程序无须任何处理就能直接在计算机上运行。但是这样的编程方式太烦琐,极易出错,效率极低,是非常不可取的。在计算机发展的早期,人们就在努力设法改变这种编程方式。开始时倾向于准备好一个由一些常用的例行程序组成的库,并借用一些代码来引用该库中的例行程序。后来改用一些字符或语言来表示这些代码,这样就成了符号语言的雏型。在此基础上,人们努力使机器语言符号化。机器语言发展成了汇编语言。语言的这一发展导致要求有一翻译程序把汇编语言程序翻译成机器语言程序,这种翻译程序称为汇编程序。

紧随汇编语言和汇编程序之后发展的是**自动编译器**。在自动编译器中,程序人员用的语言更接近通常的数学表示体系。但是用现在的标准来衡量,20 世纪 50 年代初出现的第一批自动编译器都十分初步,它们只允许简单的单目运算,数据元素的命名方式有很多限制,然而它们促进了对高级语言处理系统和通用的翻译过程的研究。

20 世纪 50 年代中期出现了 FORTRAN 等一批高级语言,与此相适应的语言处理程序、解释程序和编译程序也相继开发成功。

随着编译技术的进步和社会对编译程序需求的不断增长,50 年代末有人开始研究编译程序的自动生成工具,提出并研制**编译程序的编译程序**,它的功能是从任一语言的词法规则、语法规则和语义解释出发,自动产生该语言的编译程序。研制一个功能完全且实用的编译程序的编译程序是很困难的。多数编译程序的编译程序都是一些专用编译程序生成系统,如自动生成词法分析程序的扫描程序生成系统,自动生成语法分析程序的语法分析程序生成系统。

60 年代起,不断有人开始使用自展技术来构造编译程序。自展的主要特征是用被编译的语言来书写该语言自身的编译程序。自展的思想最早在 50 年代中间就有人提出,到 1971 年,PASCAL 的编译程序用自展技术生成后,其影响越来越大。

随着并行技术和并行语言的发展,处理并行语言的并行编译技术正在深入研究之中,将串行程序转换成并行程序的自动并行编译技术也正在深入研究之中。

分　类

按照处理方法,语言处理系统可分为编译型、解释型和混合型三类。

编译型语言处理系统是采用编译方法的语言处理系统。解释型语言处理系统是采用解释方法的语言处理系统。混合型语言处理系统是兼有编译和解释两种方法的语言处理系统。

多数高级语言都有一些不能在编译时刻确定而要到运行时刻才能确定的特征。因此,与这些特征相关联的语言成分等价的目标代码在编译时刻不能全部生成,而要到运行时刻才能全部生成。这些语言成分只能采用解释方法处理。多数解释程序都是先对源程序进行处理,把它转换成某种中间形式,然后对中间形式的代码进行解释,而不是直接对源程序进行解释。这就是说,多数高级语言处理系统既非纯编译型,也非纯解释型,而是编译和解释混合型。

基本内容

程序设计语言处理系统主要包括正文编辑程序、宏加工程序、编译程序、汇编程序、解释程序、连接编辑程序、装入程序、编译程序的编译程序、自编译程序、交叉编译程序和并行编译程序等。

正文**编辑程序**用于创建和修改源程序正文文件。一个源程序正文可以编辑成一个文件,也可以分成多个模块编辑成若干个文件。用户可以使用各种编辑命令通过键盘、鼠标器等输入设备输入要编辑的元素或选择要编辑的文件,正文编辑程序根据用户的编辑命令来创建正文文件,或对文件进行各种删除、修改、移动、复制及打印等操作。

宏加工程序把源程序中的宏指令扩展成等价的预先定义的指令序列。对源程序进行编译之前应先对源程序进行宏加工。

编译程序把用高级语言书写的程序翻译成等价的机器语言程序或汇编语言程序。编译过程可分为分析和综合两个部分。分析部分包括词法分析、语法分析和语义分析三步。分析的目的是检查源程序的语法和语义的正确性,并建立符号表、常数表和中间语言程序等数据对象。综合部分包括存储分配、代码优化和代码生成等步。综合的目的是为源程序中的常数、变量、数组等各种数据对象

分配存储空间，并将分析的结果综合成可高效运行的目标程序。

汇编程序把用汇编语言书写的程序翻译成等价的机器语言程序。

解释程序按源程序中语句的动态执行顺序，从头开始，翻译一句执行一句，再翻译一句再执行一句，直至程序执行终止。和编译方法根本不同的是，解释方法是边翻译边执行，翻译和执行是交叉在一起的，而编译方法却把翻译和执行截然分开，先把源程序翻译成等价的机器语言程序，这段时间称为**编译时刻**，然后再执行翻译成的目标程序，这段时间称为**运行时刻**。正因为解释程序是边翻译边执行，所以要把源程序及其所处理的数据一起交给解释程序进行处理。

编译方法和解释方法各有优缺点。编译方法的最大优点是执行效率高，缺点是运行时不能与用户进行交互，因此比较适用于写规模较大或运行时间较长或要求运行效率较高的程序的语言，更适用于写机器或系统软件和支撑软件的语言。解释方法的优点是解释执行时能方便地实现与用户进行交互，缺点是执行效率低，因此比较适用于交互式语言。

连接编辑程序将多个分别编译或汇编过的目标程序段组合成一个完整的目标程序。组合成的目标程序可以是能直接执行的二进制程序，也可以是要再定位的二进制程序。

装入程序将保存在外存介质上的目标程序以适于执行的形式装入内存并启动执行。

编译程序的编译程序是产生编译程序的编译程序。它接受用某种适当的表示体系描述的某一语言类中任一语言 A 的词法规则、语法规则、语义规则和（或）代码生成规则，并从这些描述产生出用目标语言 B 写的关于语言 A 的全部或部分编译程序。这样便可显著提高编译程序的开发效率。

自编译程序是用被编译的语言即源语言自身来书写的编译程序。利用自编译技术，可以从一具有自编译能力的语言 L 的一个足够小的子集 L_0 的编译程序出发，逐步构造出 L 的编译程序，也可从 L 的未优化的编译程序出发，构造优化的编译程序。

交叉编译程序是一种编译程序，它自身在甲机器上运行，生成的目标代码是乙机器的代码。

并行编译程序是并行语言的编译程序，或是将串行语言程序并行化的编译程序，后者又称为自动并行编译程序。

一个程序特别是中、大规模的程序难免没有错误。发现并排除源程序中的错误是语言处理系统的任务之一。通常源程序的语法错误和静态语义错误都是由编译程序或解释程序来发现的。排错能力的大小是评价编译程序和解释程序优劣的重要标志之一。源程序中的动态语义错误通常要借助于在语言中加入某些排错设施如跟踪、截断来发现和排除。处理排错设施的程序是**排错程序**。

展　望

语言处理系统的发展与软件语言、软件工程和软件技术的发展紧密相连，相互影响，相互促进。随着软件语言和软件技术向可视化、多媒体、并行化、智能化、自然化和自动化等方面发展，语言处理系统也向着这些方面发展。

参考文献

1. 徐家福. 系统程序设计语言. 北京：科学出版社，1983

2. Aho A V, Sethi R, Ullman J D. Compilers principles, techniques and tools. Addison-Wesley, 1986

3. Lowry M R. Software engineering in the twenty-first century. AI Magazine Fall, 1992

4. 徐家福. 软件技术漫谈. 计算机科学，1992，19(1)

（徐永森　张素琴）

yuyan zhishiku

语言知识库（language knowledge base） **自然语言处理**系统中为实现自然语言的自动分析和生成所配备的有关语言的各种知识的集合。自然语言的自动分析相当于让计算机“读”人类的自然语言，自动生成则是让计算机“写”人类的自然语言，它们是自然语言处理系统的两大主要任务。语言知识库是自然语言处理系统不可或缺的组成部分。语言知识库的规模和质量在很大程度上决定了自然语言处理系统的成败。这些语言知识附着在不同的语言单位上，如语素、词、短语、句子和篇章，涉及语言的形态、语音、词法、句法、语义乃至语用等不同层次。计算机自动处理自然语言的第一道障碍就是自然语言的潜在歧义。当程序分析“白天鹅”这个字符串时，就会产生歧义：“白　天鹅”还是“白天　鹅”？当“白天鹅”这 3 个字出现在较大的语境中，例如在句子“白天鹅飞过来了”和“白天鹅可以看家”中，这个歧义就有可能消解：前一句的正确切分是“白 天鹅 飞 过来 了”，而后一句则是“白天 鹅 可以 看家”。消解歧义有各种方法，前提是在计算机中储备一个

知识库而且程序能够从中检索到相关知识并加以运用,这个知识库以某种形式包含“白、白天、鹅、飞、过来、看家、可以、了、天鹅”这些词语,还包含“天鹅是飞禽”、“天鹅会游泳”、“鹅会游泳”、“鹅是不会飞的家禽,能看家”等常识。

语言知识库可分为两种不同的类型。一类是词典、规则库和语义概念库等,其中的知识表示是显性的,采用形式化的结构(例如:词典可采用关**系数据库**结构,规则库可采用“条件-动作”表达式),便于自动处理程序应用,但其罗列的知识会给自动处理带来歧解。例如要分析“白天鹅在湖里游泳”这句话,又会产生歧解:“白 天鹅 在 湖 里 游泳”还是“白天 鹅 在 湖 里 游泳”?因为知识库里既有“天鹅会游泳”的知识,也有“鹅会游泳”的知识。另一类知识存在于语料库之中,每个语言单位的出现,其范畴、意义、用法都是确定的。不过语料库的主体是文本,即语句的集合,每个语句都是线性的非结构化的文字序列,其中包含的知识都是隐性的。语料加工的目的就是要把隐性的知识显性化,便于机器学习和引用。词语切分将文本的汉字串改造成词语串,词的知识便显性化了,词性标注使得词类知识显性化,义项标注又使词汇语义知识显性化,语义角色标注则更进一步让句法语义知识显性化。至于建构一个具体的语言知识库,究竟要吸纳哪些语言知识,则取决于应用目标,如果要用于**机器翻译**,则还需要配备双语言的对译与转换的知识。

自然语言处理研究经过数十年的发展和积累,现在已经有了一批投入实际应用的语言知识库。英语的最多,如 WordNet,CYC,FrameNet,Treebank,PropBank 等等;日语的有 EDR 电子辞书;支持以汉语为核心的语言信息处理研究与开发的语言知识库成长迅速,其中影响广泛的有北京大学研制的综合型语言知识库以及知网 HowNet 等。中国中文信息学会建立的中文语言资源联盟为推动语言知识库的共享和发展发挥了积极的作用。

以上所列举的语言知识库都是通用型的,一般包含语言本身的知识以及客观世界的常识性知识。正如一个有用的人才除了需要具备语言知识和常识知识外,还需要具备某种专业知识一样,要使自然语言处理系统在某个专业领域实际发挥作用,也必须给它配备专业知识,需要建立领域知识库。领域知识主要包括领域内概念、概念之间的关系以及概念的属性信息等。建立领域知识库所涉及的研究内容除了**知识表示**、**知识获取**等**知识工程**的共性问题外,还要特别关注领域知识工程的特殊问题。在知识表示方面,当前领域知识库通常用领域本体(Ontology)来表示。本体具有使知识概念化、明确化、形式化以便于共享和重用等的作用。在领域知识的获取方面,通常首先要研究如何复用百科全书、教科书、专业文献等已有的知识资源,再进一步研究获取领域新知识,特别是要研究如何从**互联网**获取鲜活的可靠的知识,既可以采用人工方法,也可以采用自动化方法,更多的是采用人机互助的方法,特别要重视发挥领域专家的作用。在领域知识库支持自然语言处理技术发展的同时,已有的自然语言处理技术,如词语切分、新词发现、术语提取、术语间概念关系提取、概念层级结构自动生成等也在支持领域知识库的构建。

参考文献

1. 宗成庆,曹右琦,俞士汶. 中文信息处理 60 年. 语言文字应用,2009,4:53-61

2. 俞士汶. 建设综合型语言知识库的理念与成果的价值. 中文信息学报,2007,6,3-12

3. 董振东,董强. 知网. 见:http://www.keenage.com

4. 冯志伟. 现代术语学引论. 北京:语文出版社,1997

(俞士汶 穗志方)

yuyi

语义(semantics) 语言成分的固有含义,亦即与言语情景(参见**语用**)无关的含义。在程序设计语言中,语言成分的语义就是该语言成分在程序执行中应起之作用。语义研究涉及的理论、原则、方法以及技术所形成的学科(或谓以语义为研究对象的学科)称为语义学。

语义具有如下基本特性:

固有性 语义反映的含义是固有的,即与言语情景无关。

静态性 语言成分的语义一般是在编译时刻可以确定的。

一元性 语义可视为语言成分的一元函数。

语言成分的语义一般是用自然语言刻画的,用数学方法特别是用形式体系刻画的语义称为形式语义。随着刻画方法之不同,又可分为操作语义、指称语义、公理语义、代数语义等。

参考文献

Crice H P. Studies in the ways of words. Cambridge MA:Harvard University Press,1991

(徐家福)

yuyi Web de luoji jichu

语义 Web 的逻辑基础(logic foundation of Semantic Web)　语义 Web 体系架构的逻辑理论与逻辑语言。语义 Web 体系架构中的核心层——本体层是以**描述逻辑**为逻辑基础的。采用描述逻辑作为语义 Web 的本体语言的主要原因是:描述逻辑能够为语义 Web 提供对其至关重要的高质量的本体,而且本体的构建、整合和进化很大程度上依赖于描述逻辑的明确语义和强大推理功能。

毋庸置疑,互联网已成为交流和信息共享的重要平台,并且延伸到人类社会生活中,扮演越来越重要的角色。然而,互联网信息的机器不可解释性和不易处理性,将解读信息的繁重任务推给了人类。逻辑作为机器良性解释的一种**知识表示**和推理手段,在此恰当的时机介入互联网。它通过形式化描述网页的语义内容,让机器自动化处理成为可能,开启了语义 Web 研究。

自万维网之父 Tim Berners-Lee 先生于 2001 年提出语义 Web 理念至今,越来越多的逻辑学者投入到完善语义 Web 逻辑基础的工作中来。引入语义 Web 体系中较底层的是主谓宾三元组断言这种最为简单的逻辑形式,它与传统意义上的属性-值对相照应。为了描述断言,以及断言和其他资源间的关系,RDF(resource description framework) Schema 被引入语义 Web 体系。它类似于知识表示中的框架系统,提供父子类、父子属性的层次刻画能力,以及对属性定义域和值域的定义。为了扩展 RDF Schema 的表达能力,美国和欧洲的研究工作组同时开始着手更强的本体模型化语言计划,分别制定了 DAML-ONT 和 OIL 语言,并最终整合为 DAML + OIL。DAML + OIL 实际上是描述逻辑的一个句法变种,成为 2004 年 2 月 W3C 组织发布 Web 本体语言(OWL)的工作基础。确切地说,OWL 语言与描述逻辑中的 SHOIN(D)表达能力相当。OWL 已经能够表达传递角色 S、角色层次关系 H、枚举类型 O、逆属性 I、数量约束 N 和数据类型属性的使用(D)。

2009 年 10 月,W3C 组织修订了 Web 本体语言,制定了 OWL2。OWL2 中的 OWL2 Full 是 OWL2 DL 加上 RDF Schema 的全部。而其中的 OWL2 DL 完全就是 Web 本体语言的描述逻辑版本,与描述逻辑的 SROIQ(D)语言相照应。这一修订源于对描述逻辑 SHOIN(D)表达能力的扩展,通过增加限定性复杂角色包含公理、反身性和非反身性、角色的不相交(前三者合称 R),以及限定数量约束 Q,进一步增强描述逻辑的表达能力。OWL 与描述逻辑的紧密联系同样反映在一阶逻辑的语义解释上,部分照应关系如下表所示:

表 1　OWL、一阶逻辑语言(FOL)、描述逻辑(DL)的部分对应关系

Feature	Related OWL vocabulary	FOL	DL
全局/底层概念	owl : Thing/owl : Nothing	(axiomatize)	$\top/\bot$
交集	owl : intersectionOf	$\wedge$	
并集	owl : unionOf	$\vee$	
补	owl : complementOf	$\neg$	$\neg$
存在	owl : someValueForm	$\exists x. R(y,x) \wedge C(x)$	$\exists R.C$
任意	owl : allValueForm	$\forall x. R(y,x) \rightarrow C(x)$	$\forall R.C$
集的势的最小值	owl : minQualified-Cardinality owl : onClass	$\exists y_1,\cdots,y_n. R(x,y_1) \wedge \cdots \wedge R(x,y_n) \bigwedge_{i<j} y_i \neq y_j$	$\geqslant n\ S.C$
集的势的最大值	owl : maxQualified-Cardinality owl : onClass	$\forall y_1,\cdots,y_{n+1}. R(x,y_1) \wedge \cdots \wedge R(x,y_{n+1}) \rightarrow \bigvee_{i<j} y_i = y_j$	$\leqslant n\ S.C$
自身映射	owl : hasSelf	R(x, x)	$\exists R.Self$

描述逻辑作为一阶逻辑的一个子系统,不但提供了比一阶逻辑更清晰的表述形式,而且还保留了逻辑的可判定性。用于求解描述逻辑中推理问题的 Tableau 演算方法,被推广到语义 Web 环境下,涌现出一批卓有成效的自动机。其中代表性的有马里兰大学的 Pellet、康科迪亚大学和汉堡-哈尔工业大学的 Racer、牛津大学的 FaCT + +、阿伯丁大学的 TrOWL 等等。卡尔斯鲁厄大学开发的 KAON 是较

另类的推理机,它采用了基于归结方法的 Datalog 系统实现语义 Web 推理。

除了描述逻辑自身发展影响着语义 Web 的逻辑基础,其他逻辑的介入融合也推动着语义 Web 的发展。在工业界,以规则为逻辑基础的规则交换格式(RIF)正在试图成为 W3C 的语义 Web 标准。在学术界,**模糊逻辑**、概率逻辑、动态逻辑等,也在尝试通过与描述逻辑的混合来融入语义 Web,构建新的语义 Web 逻辑基础。

参考文献

1. 高志强, 潘越, 马力, 黄智生. 语义 Web 原理及应用. 北京: 机械工业出版社, 2009

2. Baader F, Horrocks I, Sattler U. Description logics as ontology languages for the semantic web. Mechanizing Mathematical Reasoning, 2005,228-248

(欧阳丹彤 叶育鑫)

yuyi wangluo biaoshi

语义网络表示(semantic network representation) 一种将知识中的对象与关系看作结点与有向弧,并将其连成网状的表示形式。语义网络与**框架表示**是**人工智能**领域经常使用的两种表示方法。

1968 年 Quillian 将英语词汇的语义关系表示为一种有向图结构,并称其为语义网络。在这类图中,结点与结点之间可以定义某种关系。例如,折叠椅类是椅子类的一个子类,一个具体的折叠椅是折叠椅类的一个实例,“子类”与“是一个实例”就表示了这些椅子之间的关系。进而,后人根据语义网络的原理归纳出两种主要关系:一种是“A 是 B 的子类(A-Kind-Of,缩写为 AKO)”,另一种为“A 是 B 的一个实例(IS-A,缩写为 ISA)”,这两种关系有一个重要的特性,称为继承性。所谓继承性就是 A 将继承 B 的所有特性,例如,椅子有四条腿(B),折叠椅是椅子的一个子类(A),所以它将继承椅子的这个特性,说明折叠椅也有四条腿。继承性是语义网络的最重要的性质,这个思想是计算机科学中一种重要的程序设计方法——**面向对象程序设计**的理论基础之一。

一般一个语义网络由一些有向图表示的三元组(结点 1,弧,结点 2)连结而成,其中弧是有方向的,体现了主次关系,结点 1 为主,结点 2 为辅,弧上的标注表示一个结点的属性或两个结点间的一种关系,如 ISA、A-Kind-Of(或 Part Of)等。另外,框架表示是语义网络表示一般化的表示方法,两者没有本质上的区别。

(童频 王珏)

yuyin shuru

语音输入(voice input) 用户将语音(“说话”)信号通过麦克风输入计算机,计算机将语音信号识别为文字的计算机输入方式。语音识别技术是一个典型的多学科交叉的前沿技术,涉及声学、生理学、心理学、信号处理、**模式识别**、**人工智能**、信息理论、语言学以及计算机科学等众多学科。语音输入有广泛的应用领域,如:语音听写机,声控系统等。

系统组成 大词汇量语音识别系统多采用统计模式识别技术。典型的基于统计模式识别方法的语音识别系统由以下几个基本模块构成:

(1) 信号处理及特征提取模块 主要任务是从输入信号中提取特征,供声学模型处理,目前常用的语音声学特征参数有 Mel 倒谱系数(Mel-frequency cepstral coefficients,MFCC)和感知线性预测(perceptual linear predictive,PLP)等。同时,该模块一般还采用一些信号处理技术,以尽可能降低环境噪声、信道、说话人等因素对特征造成的影响。

(2) 统计声学模型 典型系统多采用基于隐马尔科夫模型(HMM)进行建模。语音识别中使用 HMM 通常是用从左向右单向、带自环、带跨越的拓扑结构来对识别基元建模,一个音素就是一个三至五状态的 HMM,一个词就是构成词的多个音素的 HMM 串行起来构成的 HMM,而连续语音识别的整个模型就是词和静音组合起来的 HMM。

(3) 发音词典 包含系统所能处理的词汇集及其发音。发音词典实际提供了声学模型建模单元与语言模型建模单元间的映射。

(4) 语言模型 对语音识别系统所针对的语言进行建模。语言模型主要分为规则模型和统计模型两种,语言模型性能通常用交叉熵和复杂度(perplexity)来衡量。理论上,包括正则语言和上下文无关文法在内的各种语言模型都可以作为语言模型,但目前各种系统普遍采用的还是基于统计的 N 元文法及其变体。

(5) 解码器 语音识别系统的核心之一,其任务是对输入的信号,根据声学、语言模型及词典,搜索能够以最大概率输出该信号的词串。

系统实现 语音识别系统的识别基元通常采用上下文相关的音素,汉语语音识别基元也经常选择声韵母或者音节。语音识别系统所需的训练数据大

小与模型复杂度有关。大词汇量和非特定人的连续语音识别系统通常又称为听写机。其架构就是建立在前述声学模型和语言模型基础上的 HMM 拓扑结构。训练时对每个语音识别基元用前向后向算法获得 HMM 模型参数，识别时，将基元串接成词，词间加上静音模型并引入语言模型作为词间转移概率，形成循环结构，用 Viterbi 算法进行解码。针对汉语易于分割的特点，先进行分割再对每一段进行解码，可以有效地提高解码的效率。

自适应与强健性 语音识别系统的性能受许多因素的影响，包括不同的说话人、说话方式、环境噪声和传输信道等。提高系统强健性，就是使系统在不同的应用环境和条件下保持性能的稳定性。语音识别系统自适应的目的，就是根据不同的影响来源，自动地、有针对性地对系统进行调整，在使用中逐步提高语音识别系统性能。 （陶建华）

yuyong

语用（pragmatics） 语言成分相对言语情景的含义。言语情景包括发信人、收信人、语境、目标等。对**程序设计语言**来说，语用乃语言成分在程序特定执行中的实际效用。语用研究涉及的理论、原则、方法以及技术所形成的学科（或称以语用为研究对象的学科）称为语用学。

语用具有如下基本特性：

相对性 语用反映的含义是相对言语情景而言的，不是语言成分的固有含义。

动态性 语言成分相对言语情景而言的含义可视为语言成分的固有含义的实例，它是动态生成的。

多元性 语用可视为语言成分与言语情景的二元函数。

今举二例如下：

例 1 What did you mean by X?

这里反映的 X 的含义是语用，它是 X 和 you 的二元函数。

例 2 Ada 程序段。

```
pragma …… ; Csi;……
    opt …… ;Csj;…… Opt
pragma
```

其中 Csi，Csj 均为循环语句，其含义和言语情景有关，Csi 除其循环语句的含义外，还告知编译程序，其目标码无须优化；而 Csj 则除其循环语句的含义外，还告知编译程序其目标码需优化。因而，这里 Csi、Csj 的含义均为语用。

参考文献

Geoffrey N L. Principles of pragmatics. Longman, 1983

（徐家福）

yu

域（field） 一种具有两个二元运算的代数结构。

如果至少含有两个元素的集合 F 及其上的两个二元运算加法“+”和乘法“*”满足以下公理：

Ⅰ$_1$ 加法交换律 对任意 $a,b\in F$ 有 $a+b=b+a$；

Ⅰ$_2$ 加法结合律 对任意 $a,b,c\in F$ 有 $a+(b+c)=(a+b)+c$；

Ⅰ$_3$ 存在零元 存在 $0\in F$，使得对任意 $a\in F$ 有 $a+0=a$；

Ⅰ$_4$ 存在负元 对任意 $a\in F$，有 $-a\in F$ 使 $a+(-a)=0$；

Ⅱ$_1$ 乘法交换律 对任意 $a,b\in F$ 有 $ab=ba$；

Ⅱ$_2$ 乘法结合律 对任意 $a,b,c\in F$ 有 $a(bc)=(ab)c$；

Ⅱ$_3$ 存在幺元 存在 $e\in F$，使得对任意 $a\in F$ 有 $ae=a$；

Ⅱ$_4$ 存在逆元 对任意 $a\in F$，有 $a^{-1}\in F$ 使 $aa^{-1}=e$；

Ⅲ 乘法对加法的分配律 对任意 $a,b,c\in F$ 有 $a(b+c)=ab+bc$ 和 $(a+b)c=ac+bc$；

则称 F 为一个域。域是交换环且无零因子。

全体有理数、实数和复数关于普通的加法和乘法都构成域，分别称为有理数域、实数域和复数域。

设 p 为一个素数。模 p 的全体剩余类 $\{\overline{0},\overline{1},\cdots,\overline{p-1}\}$ 关于模 p 的加法 $+_p$ 和模 p 的乘法 $*_p$ 构成一个域，记为 $\mathbf{Z}/(p)$。

若 F 和 F' 为两个域，则环同态（同构）f: $F\to F'$ 就是域同态（同构）f: $F\to F'$。域 F 关于它的理想 D（即 D 是环 F 的理想）的商环 F/D 是域，当且仅当 D 是 F 的极大理想。

如果域 E 的子集 F 关于 E 的运算也构成域，则称 F 为 E 的**子域**，E 为 F 的**扩域**。有理数域是实数域的子域，实数域是有理数域的扩域。

设 F 为一个域。若对任意正整数 n 有 $ne\neq 0$，则称 F 的特征为零；否则，有最小正整数 p 使 $pe=0$，这时称 F 的特征为 p。显然 p 必是素数，而且对任意 $a,b\in F$ 有 $(a+b)^p=a^p+b^p$。

如果域 F 没有真子域即不同于 F 的子域，则称

F 为**素域**。若素域 F 的特征为零，则 F 与有理数域同构；若素域 F 的特征为 p，则 F 与域 $\mathbf{Z}/(p)$ 同构。

研究域 E 的一般方法是取定 E 的一个子域（如 E 的素子域）F 作为基域，然后讨论 E 相对 F 的代数性质。对 E 的任意子集 M，用 $F(M)$ 表示由 F 和 M 中元素经有理运算（加、减、乘、除）得到的所有元素的集合，则 $F(M)$ 既是 E 的子域，又是 F 的扩域，称为由 M 产生的 E,F 的中间域。若 $M=\{\alpha_1,\alpha_2,\cdots,\alpha_n\}$，则 $F(M)$ 记为 $F(\alpha_1,\alpha_2,\cdots,\alpha_n)$。称 E 的子域 $F(\alpha)$ 为 F 的**单扩张**。F 的任意扩张都可以通过一系列单扩张得到。

设 E 是 F 的扩域。若 E 中每个元素都是 F 上某多项式的根，则称 E 为 F 的**代数扩张**，否则称 E 为 F 的**超越扩张**。如果 F 上每个多项式的根都在 E 中，则称 E 是 F 的**代数闭域**。有理数域的代数闭域是代数数域，实数域的代数闭域是复数域，域 $\mathbf{Z}/(p)$ 的代数闭域是多项式的分裂域。

如果 F 为有穷集，则称域 F 为**有限域**或**伽罗瓦域**。最简单的有限域是 $\mathbf{Z}/(p)$，它的特征是 p。取 $\mathbf{Z}/(p)$ 上一个 n 次不约多项式 $f(x)$，做 $f(x)$ 关于 $\mathbf{Z}/(p)$ 的分裂域即 $\mathbf{Z}/(p)$ 的包含 $f(x)$ 的所有根的最小代数扩域，这个域恰有 p^n 个元素。因为元素个数相同的有限域都互相同构，所以有限域的元素个数只能有 p^n 的形式，其中 p 为素数且 n 为正整数。具有 p^n 个元素的有限域记作 $GF(p^n)$。它是 $\mathbf{Z}/(p)$ 关于某个不可约多项式的分裂域。

域，特别是有限域，在理论上和算法上与计算机科学的许多领域密切相关，例如算法理论、复杂性理论、编码理论和机械定理证明等。另外，关于域和有限域的一些计算机方法的研究也在迅速发展。

参考文献

1. 万哲先. 代数和编码. 北京：科学出版社，1976

2. 熊全淹. 近世代数. 上海：上海科技出版社，1978

3. 王兵山，周贤林，王长英，何自强. 离散数学. 长沙：国防科大出版社，1985

（李廉　王兵山）

yuming xitong

域名系统（domain name system，DNS） 互联网中实现网络应用名的分级管理和使用机制。DNS 有两个概念上独立的要点：一个是抽象的，用来指明名字语法和名字的授权管理规则；另一个是具体的，用来指明一个分布式计算系统的层次结构，并能高效地将名字映射到 IP 地址。

语法上，每个组织或网络应用的域名由一系列字母和数字构成的段组成，每个段称为一个标签，用小数点分隔。域名是有层次的，按其表示方式，最右一个标签为最高部分，称为顶级域名，包括国家域名或通用域名，最左边的部分处于最低层次。

域名和 IP 地址（参见**网际协议**）是一一对应的，域名易于记忆，用得更普遍。当用户要和 Internet 上某个网络应用交换信息时，只需使用域名，网络会自动转换成 IP 地址，找到该应用所在的那台计算机。

域名系统规定了顶级域的值，每个顶级域可以定义其下二级域的值，并可依次获得第三级域、第四级域等的值。当一个组织希望参加域名系统时，必须在某一级域下申请一个域名。一旦一个组织拥有一个已申请的域，就可以决定是否设置进一步的层次结构。对于一个规模小的组织可以不再设置下一级的域，而对于规模大的组织就有必要设置多层结构。

由于域名是一个逻辑概念，所以它与所处的物理位置可以没有关系。而且对一个组织来说有不同的部门，可以根据部门的实际情况选择不同层次的域名构造。

域名系统的一个特点是自治，即允许每个组织为其应用设置域名或改变这些域名。大多数具有 Internet 连接的组织都运行一个**域名服务器**，称 DNS 服务器。每当应用需要将域名翻译为 IP 地址时，应用成为域名系统的一个客户。这个客户将待翻译的域名放在一个 DNS 请求信息中，并将这个请求发给 DNS 服务器。服务器从请求中取出域名，将其翻译为对应的 IP 地址，然后，用一个回答信息将结果地址返回给应用。

域名对地址的映射由一组域名服务器完成。域名服务器中包含提供域名对地址转换，域名对 IP 地址映射的服务程序。

图 1 是一个树结构的域名服务器的概念布局。树的根是能够识别顶级域，并知道如何解析每个域名的服务器。给定要解析的名字后，根可为该名字选择一个正确的服务器，并逐级往下搜索，最后将结果返回。

概念树中的链接并不表示物理网的连接，而是解析域名的一种逻辑连接。服务器树是用于 Internet 通信的一种抽象结构。

从概念上讲，域名转换自上而下进行，从根服务器开始，逐级处理，直到树叶上的服务器。客户机中

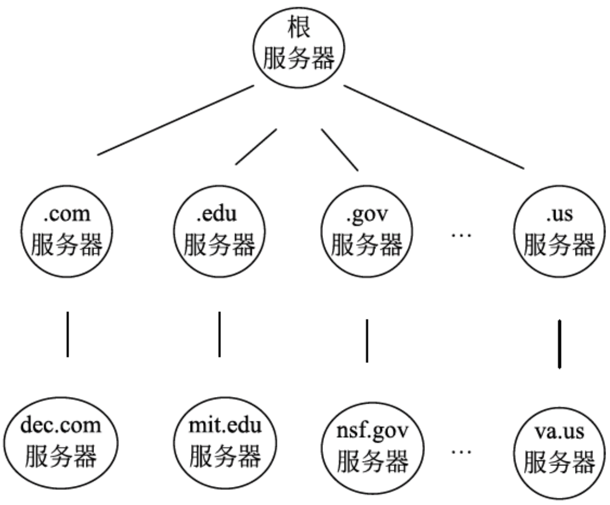


图 1 域名服务器树结构

配置有 DNS 解析器,它与指定的本地域名服务器联系,而本地的域名服务器必须知道根服务器地址,以便按如图 1 所示的方式解析域名。域名服务器使用指定的 UDP 端口(53)通信,以便客户机方便地与域名服务器通信。

参考文献

1. 胡道元. 计算机网络(高级). 北京: 清华大学出版社, 1999

2. Comer DE. Internetworking with TCP/IP, Volume Ⅰ. 4th ed. Englewood Cliffs, NJ: Prentice Hall, 2000

(胡道元)

yuming xitong anquan kuozhan

域名系统安全扩展(domain name system security extensions, DNSSEC) 域名系统(DNS)扩展的一组安全机制,主要依靠数字签名机制保护域名系统信息的真实性、完整性,使域名解析服务器可以验证它所收到的应答是否来自于真实的服务器,或者是否在传输过程中被篡改过。DNSSEC 主要为了防范针对 DNS 系统缓存污染(cache poisoning)类型的攻击,它不保护 DNS 数据传输的保密性,也不能用于防范拒绝服务等其他类型的攻击。

DNS 系统的安全漏洞早在 20 世纪 90 年代初就引起了广泛关注,第一份关于 DNSSEC 的标准 RFC2065 颁布于 1997 年,之后不断改进的标准包括 RFC2535(1999),RFC4033、4034 和 4035(2005),以及之后的 RFC5155 等一些补充性标准。

为了实现资源记录的签名和验证,DNSSEC 增加了四种类型的资源记录:RRSIG(resource record signature)、DNSKEY(DNS public key)、DS(delegation signer)、NSEC(next secure)。

DNSSEC 对域中的每个资源记录集合(RRSet)进行数字签名,并将其存储在对应的 RRSIG 记录中。DNSKEY 记录则保存了签名私钥所对应的公钥,供验证签名的有效性所用。这个公钥的哈希值保存在上级域名服务器的 DS 记录中,从而在域名的层次结构中形成信任链。在 DNSSEC 没有全面部署的情况下,可以使用一些信任锚(trust anchor,即可信的域名服务器的 DNSKEY 或 DS)来建立信任关系。

DNSSEC 要求域名文件中的所有 RRSet 记录按字母顺序排列,并使用 NSEC 记录将这些 RRSet 记录隔开,每个 NSEC 记录指出排序后的下一个资源记录的名称以及这一名称所有资源记录类型,使 DNSSEC 可以检测那些不存在的资源记录。

除新增的资源记录以外,DNSSEC 要求必须支持 EDNS0(RFC2671,即允许 DNS 报文大小必须达到 1220 字节,而不是最初的 512 字节),同时在 DNS 报文头中增加了三个标志位:DO(DNSSEC OK),AD(authentic data),CD(checking disabled)。

目前主流的域名服务软件(如 BIND)的实现均已支持 DNSSEC 标准,但是大规模部署从工程角度还存在许多问题,包括使用和维护方面。在 ICANN 和 VeriSign 推动下,2010 年 7 月所有 13 个根域名服务器完成了 DNSSEC 部署,顶级域名也在计划之中。一些大的 ISP 已经开始部署支持 DNSSEC 的解析服务器。尽管 DNSSEC 不仅对于域名系统非常重要,还为大规模的密钥管理提供了新的可能,可望为电子邮件、Web 系统的安全提供支持,但是由于它的使用和维护要比传统的 DNS 复杂得多,因此它离全面部署还有很长的路要走。

参考文献

1. Arends R, et al. RFC 4033: DNS security introduction and requirements. 2005

2. Arends R, et al. RFC 4034: Resource records for the DNS security extensions. 2005

3. Arends R, et al. RFC 4035: Protocol modifications for the DNS security extensions. 2005

(段海新)

yuan qifashi sousuo

元启发式搜索(metaheuristic search) 是一类搜索算法。它精心设计了高层搜索策略和局部搜

索过程之间的交互,以构建一个具有跳出局部最优解的能力且能在问题空间中进行健壮搜索的过程。它是自20世纪80年代初以来,研究者基于客观世界中的自然现象提出来的具有一定普适性的**启发式搜索**算法,在一些文献中也被称为现代启发式算法。主要包括禁忌搜索算法(tabu search)、模拟退火算法(simulated annealing)、遗传算法(genetic algorithm)、蚁群优化算法(ant optimization algorithm)、粒子群优化算法(particle optimization algorithm)、迭代局部搜索算法(iterated local search)和人工神经网络算法(artificial neural network)等。这些搜索算法具有优化性能高、无须问题特殊信息等优点,广泛应用于计算机科学、优化调度、交通运输、工程优化设计等领域,但不具备完备性。

随着研究的发展,元启发式搜索也涵盖所有包含了克服在复杂解空间中陷入局部最优的策略的过程,特别是使用了一种或多种邻域结构来定义可接受的移动、并用这些移动将一个解转换到另一个解或以构造性和破坏性的方式来组建解或扰动解的过程。各种元启发式在算法结构、优化机制等方面存在一定的差异,但均采用局部搜索结构,即都是从一个或一组初始解出发,搜索邻域中的解,按接受准则更新当前状态并继续搜索过程。如此重复搜索步骤直到满足算法的收敛准则,得到问题的优化结果。

元启发式搜索主要用来求解复杂性不断增加、缺少形式定义的具有离散搜索空间的现实问题,特别是组合优化问题,但也用于实数搜索空间。与普通的搜索不同,元启发式搜索以一个或一组解为迭代的初始值,通过对问题的优化参数进行编码,将它们映射为可进行启发式操作的数据结构,并采用结构化和随机化的搜索策略,根据优化的目标函数值的信息——而不是它的导数信息——进行搜索。元启发式搜索具有全局优化性能高、鲁棒性强、通用性强且适于并行处理的特点,适用范围非常广泛,且算法易于修改,特别适用于大规模并行计算。

目前元启发式搜索的数学基础还不够完善,缺乏普遍适用的算法收敛性理论和完备的计算搜索效率及时空复杂度的理论评价体系。还没有建立起完备的算法结构框架,还不能清楚地解释算法在应用过程中出现的各种问题。算法参数的选取主要是凭借经验,对算法操作算子的作用机理及其作用效果的分析还不充分。此外,元启发式搜索不能保证找到最优解,甚至不能保证找到接近最优解的满意解,其寻优能力和收敛能力主要还是通过经验来评价。因此在使用元启发式搜索求解问题时,往往需要通过一些实验才能确定哪种或哪些元启发式搜索算法更适合所求解的问题。这些都是今后需要研究解决的问题。

参考文献

1. Gendreau M, Potvin J Y. Handbook of metaheuristics. 2nd ed. Springer, 2010

2. Talbi E G. Metaheuristics: From design to implementation. Wiley & Sons, 2009

3. 邢文训,谢金星. 现代优化计算方法. 2版. 北京: 清华大学出版社, 2005

(董兴业 黄厚宽)

yuanqijian ceshi

元器件测试(component testing) 用各种专用设备与有效方法,根据一定标准,对元器件进行的测试与选择。其目的是测试元器件的特性参数和性能以及元器件筛选前后特性参数和性能的变化,从而剔除不符合技术条件要求的元器件,以提高整机系统的可靠性和稳定性。

集成电路测试以直流参数、交流或动态参数、静态及动态功能测试为主。直流参数用于测量电流、电压和功耗等特性,交流参数或动态参数用于测量频率特性或脉冲特性,功能测试用于验证逻辑图形或真值表。静态功能测试在直流或低工作频率下进行,而动态功能测试则在最高工作频率下进行。线性集成电路测试以测量直流和交流参数为主。

晶体管测试通常以特性参数为主,测试项目有反向击穿电压、反向漏电流、电流放大倍数等。某些情况下,为了反映环境条件对特性参数的影响,还要进行高温或低温环境下的特性参数测试。

元件方面的测量主要是在不同的温度环境下进行静态参数的测试,如钽电解电容器通常在最高额定工作温度(85 ℃ ±2 ℃)下测试漏电流。

早期的元器件测试设备比较简单。随着电子计算机技术的飞速发展,元器件测试实现了测试过程、数据处理等自动化,测试精度也提高了。

参考文献

陈炳生. 电子可靠性工程. 北京: 国防工业出版社,1987

(于海环)

yuanqijian kekaoxing

元器件可靠性(component reliability) 在元器件寿命期内和规定的条件下,保持其规定功能的

能力。元器件可靠性分为固有可靠性和使用可靠性两类。

（1）元器件固有可靠性　由元器件的设计、材料和制造工艺等因素所决定。这些因素主要包括在批量生产中原材料选定、工艺条件设置、设备状况、人为因素的变动，以及线路、版面与结构设计的缺陷等。

（2）元器件使用可靠性　由元器件的选择与应用因素所决定。如在线路设计、布局和整机装配、调试中没能很好地考虑电和机械过应力对元器件的损伤，各种干扰和特性变化引起元器件误动作等因素都会影响元器件的可靠性。

元器件可靠性可用失效率 $\lambda(t)$ 来表征，$\lambda(t)$ 即产品在 t 时刻后的时间间隔 $[t,t+\Delta t]$ 内失效数与在 t 时刻还在正常工作的产品数之比，称为失效率函数，简称失效率。大量的使用和试验结果表明，电子元器件典型的失效率曲线如图 1 所示。显见，失效率大致可划分为三个阶段，即早期失效期、偶然失效期和耗损失效期。

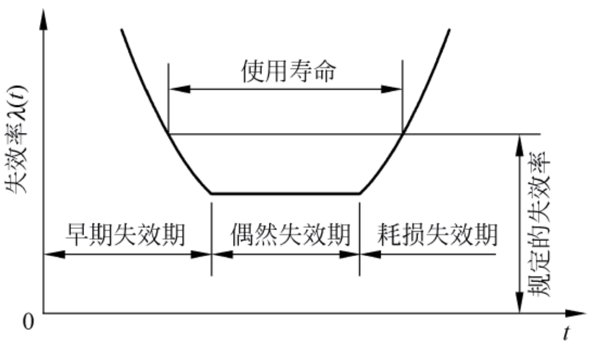


图 1　电子元器件典型的失效率曲线

早期失效期出现在元器件开始工作后的较早时期，其特点是失效率较高，且元器件失效率随时间的增加而迅速下降。早期失效是由于设计和制造工艺上的缺陷等因素造成的，可以通过加强对原材料和工艺的检验，进行筛选等办法来淘汰早期失效的元器件，从而提高元器件可靠性。偶然失效期出现在早期失效之后，其特点是失效率低且稳定，近似常数。元器件的偶然失效期是元器件可靠工作的时期，研究这一时期的失效因素有重要意义。耗损失效期出现在元器件使用的后期，其特点是失效率随时间的增加而上升。耗损失效主要是由于元器件的老化、疲劳、损耗造成的。改善耗损失效的方法是不断提高元器件的工作寿命，对寿命短的元器件，在它们到达耗损失效期前就及时予以更换。为了提高元器件的可靠性，必须掌握元器件的失效规律，并采取有效措施。

参考文献

胡昌寿. 可靠性工程——设计、试验、分析、管理. 北京：宇航出版社，1988　（于海环）

yuanqijian laohua

元器件老化（component burn-in）　元器件的缺陷绝大部分发生在开始的几小时至几十小时之内（取决于制造制程的成熟程度和器件总体结构）称为早期故障。在设定的时间间隔内对元器件连续施加恶劣的工作环境和电气性能条件，进行苛刻的筛选试验，有助于故障加速出现，使元器件的缺陷在出厂前暴露的方法和过程，称为元器件老化，从而达到剔除早期失效元器件的目的。

元器件老化是元器件筛选的一种形式。它对工艺制造过程中可能存在的一系列缺陷，如表面玷污、引线焊接不良、沟道漏电、硅片裂纹、氧化层缺陷、局部发热点、二次击穿等都有较好的筛选效果。对于无缺陷的元器件也可促使其电参数稳定。

一般计算机中需作老化的元件有电阻、电容等，器件有半导体晶体管和集成电路等。常用的老化设备有电阻、电容老化台，晶体管老化设备和**微型计算机**控制的模拟、数字电路老化系统等。

半导体器件常用的老化筛选方法如下。

常温静态功率老化　老化时，器件所处的环境温度是室温，器件加正向偏压导通，老化所需要的热应力是由器件本身所消耗的功率转换而来。

高温静态功率老化　高温静态功率老化的加电方式及试验电路的形式均与常温静态功率老化的相同。区别在于前者在较高的环境温度下进行老化，集成电路的结温很高，因此，一般而言，前者的老化效果要比后者的好。

高温反偏老化　老化时，器件被同时加上高温环境应力和反向偏压电应力，器件内部无电流或只有微小的电流通过，几乎不消耗功率。这种老化方式对剔除具有表面效应缺陷的早期失效器件特别有效，因而在一些表面敏感的半导体器件中得到广泛应用。

高温动态老化　主要用于数字电路。其老化方法是，在高温环境下，向电路的输入端输入脉冲信号，使电路不停地处于翻转状态。这种老化方式比较接近电路的实际使用状态。

计算机所用的元件（电阻、电容等）主要采用高

温静态功率老化的方法进行老化。

参考文献

胡昌寿. 可靠性工程——设计、试验、分析、管理. 北京：宇航出版社,1988 （于海环）

yuanqijian shaixuan

元器件筛选（component screening） 将不符合使用要求的元器件通过目检或施加应力等方法使其缺陷暴露进而予以剔除的过程。从广义上讲,在元器件生产过程中各种工艺质量检验以及半成品、成品的电参数测试都是筛选。通常所提的计算机元器件筛选是指元件（电阻、电容等）,器件（二极管、三极管、集成电路等）在安装使用前进行的筛选。筛选的方式有以下几种。

1. 目检筛选　根据被筛选元器件结构特点和目检要求,选用不同放大倍数的放大镜或显微镜,检查元器件的外观质量。目检筛选也可借助于红外显微镜、X 射线仪等设备对元器件的内部质量进行检查。

2. 温度应力筛选　在指定温度下进行。可分为以下三种情况：

（1）高温存储　被筛选的元器件不加电应力,温度恒定。这项试验对有表面玷污缺陷的元器件有筛选作用。

（2）高温电老化　被筛选的元器件加电应力,温度恒定。这项试验为具有加速性的筛选,它能暴露元器件的潜在缺陷,从而把早期失效的元器件剔除。

（3）温度循环　被筛选的元器件不加电应力,温度按一定规律,由低温突变为高温,随后又由高温突变为低温。这项试验对材料系数不匹配形成的缺陷或芯片已有裂纹的器件有筛选作用。

3. 机械应力筛选　包括恒定加速度试验,振动试验（包括等幅振动、恒频振动、扫描振动、随机振动等）以及早期经常采用的跌落试验等。恒定加速度筛选试验对外壳封装不良,芯片粘结不牢,金丝内引线键合力不足等缺陷有筛选作用。

为了检查有空腔的元器件内有无可动多余物,可采用颗粒碰撞噪声检测（一种特殊的机械应力筛选）。

4. 气密性筛选　采用浸在液体中检查是否漏气,放在真空环境中检查是否漏气等方法,其目的在于剔除达不到密封性要求的元器件。在选择筛选方法时要考虑元器件的封装形式和有效的内腔体积。

为了检查筛选前后元器件主要性能的变化,应对元器件的主要性能进行测试,测试的仪器设备应符合规定的精度要求。凡要求在筛选过程中进行监测的元器件,引线应尽可能短,测试过程不得产生寄生振荡。测试静电敏感器件时要采取防静电措施。

参考文献

朱明让. 质量与可靠性管理. 北京：宇航出版社,1988 （于海环）

yuanshuju guanli

元数据管理（metadata management） 对海量存储系统中描述文件的多种属性信息的元数据进行组织和管理,以利于大规模数据快速查找的技术。

虽然存储系统中元数据的数据量远小于整个系统的存储容量,但是通常情况下,元数据相关操作却占整个文件系统操作的很大比例。因此,元数据管理是存储系统性能优化的关键所在。

存储系统元数据管理经历了多个发展阶段,从集中式客户端/服务器结构、到单元数据服务器、再到多元数据服务器集群结构。在早期的存储系统中,存储的数据是由文件服务器来负责管理和维护的,文件系统同时管理和维护数据和元数据,它们在物理位置部署上尽量符合局部性特征原则,这种元数据组织模式实际上限制了文件系统的可扩展性。为提高元数据管理的性能,存储系统多采用数据和元数据分离的模式,即用存储服务器存储数据,用单独的元数据服务器来存储元数据（参见**对象存储系统**）。随着近年来存储系统容量的不断增长和数据复杂度的提高,采用单一的元数据服务器来管理整个文件系统的结构已经成为影响系统性能提高的主要问题之一。为解决性能的提高问题,可以采用多个元数据服务器,即元数据服务器集群,来提高存储系统的可扩展性和改善整个系统的实际性能。

目前,在存储系统中,元数据管理的方法主要有两种。一种是基于目录子树划分的方法,其基本思想是以子目录为单位,根据目录子树来设置元数据的分布。在实际应用中,将元数据按照目录划分到不同元数据服务器中,每个元数据服务器管理目录层次中的多个子目录树。基于目录子树划分的方法保持了目录层次结构中的局部性特性,便于在硬盘上实现高效的顺序读写操作。同时,此方法需要考虑负载均衡问题和对于热点数据的管理问题。另一种元数据管理方法是基于哈希计算的方法。这种方法采用元数据的标识或者其他相关属性作为键值来

进行哈希计算,在元数据服务器中分配名字空间。这种方法能够比较均匀地分配工作负载,实现负载均衡,便于快速查询操作等。同时,这种方法需要考虑有效的缓存和预取机制,简化有关目录的指令操作,提高系统的可扩展性。

在大规模存储系统中提供高效的元数据查询是影响系统整体性能的关键问题,相关研究内容包括基于资源(如 CPU、内存以及磁盘)最小化的查询方法、快速可扩展的索引构建、查询和更新机制以及支持多种查询访问的文件系统接口设计方法等。

参考文献

Weil S, Brandt S A, Miller E L, et al. Ceph: a scalable, high performance distributed file system. Proc OSDI, 2006

（华宇）

yuan yuyan

元语言(metalanguage) 用于描述另一语言的语言。在被描述的语言中不出现元语言的符号,元语言常用于描述**程序设计语言**的语法定义。例如,通常用巴克斯-诺尔形式体系(**BNF**)这种元语言来描述一般程序设计语言的语法。

元语言符号	意义
::=	定义为
< >	生成项
|	或

使用这些元语言符号,数字就可定义为:

<数字> ::=0|1|2|3|4|5|6|7|8|9

无正负号整数可定义为:

<无正负号整数> ::= <数字> | <无正负号整数><数字>

（程虎）

yuanshi digui hanshu

原始递归函数(primitive recursive function) 一类特殊的可计算的**数论函数**。所谓数论函数是指:自变量值和函数值都是自然数的函数。所有原始递归函数都可通过有限次应用以下规则来定义:①函数值恒为 0 的零函数 C_0、函数值为自变量值加 1 的后继函数 S 以及函数值为第 i 个自变量值的 n 元投影函数 $P_i^{(n)}$ 均是原始递归函数;②原始递归函数的合成仍是原始递归函数;③若 g 和 h 分别是 n 元和$(n+2)$元原始递归函数,则由 g 和 h 按原始递归定义的$(n+1)$元函数 f 也是原始递归函数。其中:

$$f(x_1,\cdots,x_n,0)=g(x_1,\cdots,x_n)$$

$$f(x_1,\cdots,x_n,y+1)=h(x_1,\cdots,x_n,y,f(x_1,\cdots,x_n,y))$$

显然,所有原始递归函数都是直观可计算的全函数。许多常用的全函数都是原始递归函数,特别地,所有初等函数都是原始递归函数。但并非一切直观可计算的全函数都是原始递归函数。例如:阿克曼函数就不是原始递归函数。

原始递归函数具有以下重要性质:

(1) 若数论函数 f 的函数值只对自变量值的有限个组合非零,则 f 是原始递归函数。

(2) 若 g 是$(n+1)$元原始递归函数,则由 g 按受限最小根定义的$(n+1)$元函数 f 也是原始递归函数。其中:

$$f(x_1,\cdots,x_n,y)=\mu^y z[g(x_1,\cdots,x_n,z)=0]$$

即,若存在值 z 满足 $0\leqslant z\leqslant y$,且使 $g(x_1,\cdots,x_n,z)=0$,则 $f(x_1,\cdots,x_n,y)$ 等于使 $g(x_1,\cdots,x_n,z)=0$ 的最小 z 值,否则等于 $y+1$。

(3) 若 $g_1,\cdots,g_m$ 是 n 元原始递归函数,且 $h_1,\cdots,h_m$ 是$(n+m+1)$元原始递归函数,则由 $g_1,\cdots,g_m$ 和 $h_1,\cdots,h_m$ 按联列递归定义的$(n+1)$元函数 $f_1,\cdots,f_m$ 都是原始递归函数。其中:

$$f_1(x_1,\cdots,x_n,0)=g_1(x_1,\cdots,x_n)$$

$$\vdots$$

$$f_m(x_1,\cdots,x_n,0)=g_m(x_1,\cdots,x_n)$$

$$f_1(x_1,\cdots,x_n,y+1)=h_1(x_1,\cdots,x_n,y,f_1(x_1,\cdots,x_n,y),\cdots,f_m(x_1,\cdots,x_n,y))$$

$$\vdots$$

$$f_m(x_1,\cdots,x_n,y+1)=h_m(x_1,\cdots,x_n,y,f_1(x_1,\cdots,x_n,y),\cdots,f_m(x_1,\cdots,x_n,y))$$

(4) 若 g 和 h 分别是 n 元和$(n+2)$元原始递归函数,则由 g 和 h 按值程递归定义的$(n+1)$元函数 f 也是原始递归函数。其中:

$$f(x_1,\cdots,x_n,0)=g(x_1,\cdots,x_n)$$

$$f(x_1,\cdots,x_n,y+1)=h(x_1,\cdots,x_n,y,\langle f(x_1,\cdots,x_n,0),\cdots,f(x_1,\cdots,x_n,y)\rangle)$$

这里$\langle f(x_1,\cdots,x_n,0),\cdots,f(x_1,\cdots,x_n,y)\rangle$表示序列 $f(x_1,\cdots,x_n,0),\cdots,f(x_1,\cdots,x_n,y)$ 的**哥德尔配数**。

(5) 若 g_1,g_2 和 h 分别是 1 元、1 元和 4 元原始递归函数,则由 g_1,g_2 和 h 按双重递归定义的函数 f 也是原始递归函数。其中:

$$f(0,y_2)=g_1(y_2)$$

$$f(y_1+1,0)=g_2(y_1)$$

$$f(y_1+1,y_2+1)=h(y_1,y_2,f(y_1+1,y_2),f(y_1,y_2+1))$$

(6) 有人已经证明：更一般的“范数序”意义下的函数递归定义方案也保持原始递归性。

原始递归函数和一类受限的编程语言之间存在对应关系。

参考文献

Hennie F. Introduction to computability. Boston, MA：Addison - Wesley,1977 （殷建平）

yuanxing sucheng fangfa

原型速成方法(rapid prototyping method)

一种快速建立预期系统的原型的软件开发方法。原型是预期系统的一个可执行模型,它正确地反映了系统性质的一个选定的子集(如功能、显示形式、计算结果或响应时间等)。原型可用来明确地表达和确认需求,验证所用设计方案的可行性和支持软件的演化。原型必须能快速、准确和低成本地构作和修改,并且是可运行的。

开发的原型可以在达到某种目的后被抛弃,也可以逐步将其演化成最终的软件产品。采用**抛弃策略**的原型主要用于目标软件的需求模糊不清的场合,或者难以确定设计方案可行性的场合。这类原型在明确了目标软件的需求或验证了设计方案后即被抛弃。由于原型最终被抛弃,所以,原型可以使用与目标软件不同的编程语言,可以运行于与目标软件不同的软硬件环境中。但是,它增加了对最终产品无用的原型开发费用和时间,因此,原型的开发必须是快速的和低成本的。同时这种方法还会诱使开发者省略或简化文档工作,这是有害的。抛弃策略是技术水平不够的一种权宜之计,最适合在项目早期阶段用来产生粗略的系统模型。

演化策略是在初始原型的基础上,通过不断扩充和完善(每次扩充和完善产生一个新的原型版本),直至得到最终的软件产品。这类原型通常只提供目标软件的部分功能和性能,它们将是最终软件的一部分,因此,原型的准确规约和清晰的设计文档对有效的软件原型速成是至关重要的。

图1给出了一个演化的软件原型速成过程。首先对用户的要求进行需求分析,然后设计并产生初始的原型,用户在观看了原型演示后,对原型作出评价,用户的反馈意见可能会引起新一轮的需求分析—设计原型—生成原型—演示原型的过程。原型在设计后,也可通过静态分析验证其合理性。原型的每个后续版本都应该更加接近于最终系统。被用户接受的原型经由一个可选的优化过程产生出发布的产品。

有时可以针对所需软件的各种问题开发多个小的原型,每个小原型回答一个问题,这会更有用。例如,针对用户界面的界面原型,针对某些特定功能的功能原型等。这种做法的好处在于反映系统不同方面的原型可以并行独立开发,每个原型都相对比较小、简单、容易改变。

原型速成方法已经被广泛接受。**演化模型**、**螺旋模型**和快速应用开发(RAD)等都是基于原型速成方法的。虽然开发原型需要花费一定的成本,但由于原型有利于明确需求和确定设计方案,因此,采用原型速成方法能避免因需求不清或设计方案不合理而造成的损失。

参考文献

Marciniak J J. Encyclopedia of software engineering. John Wiley & Sons Inc. , 2002 （钱乐秋）

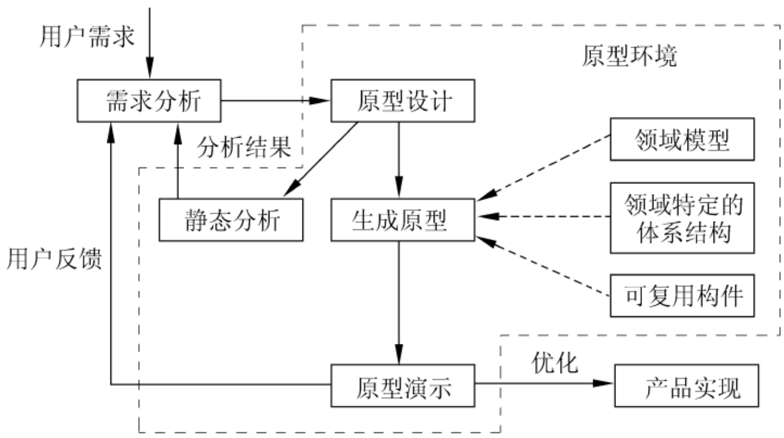


图1 原型速成过程

yuancheng guocheng diaoyong

远程过程调用(remote procedure call, RPC)　在分布式软件系统中,一种支持一台计算机上的程序像调用本地子程序一样调用另一台计算机上子程序的同步通信机制。当涉及的软件采用面向对象编程,远程过程调用也可称作远程调用或远程方法调用。

远程过程调用的典型流程如下:①客户方调用位于本地的框架函数,相关的调用参数压入堆栈中;②客户方的框架函数将所有参数打包成一条信息,这种打包过程被称作编码;③客户方通过包交换通信向服务方发送该打包信息;④服务方的框架函数接收打包信息,并解码还原打包信息;⑤服务方的框架函数调用服务方的本地过程;⑥返回信息再以同样地步骤由服务方反方向传递给客户方。这里,客户方和服务方两个框架函数类似于河流两岸的转接站,是负责发送、接收和处理相应调用信息的函数型程序。

参考文献

Sun Microsystems. RPC: Remote Procedure Call, Protocol Specification Version 2, http://tools.ietf.org/html/rfc1057

(滕猛)

yuancheng guocheng diaoyong zhongjianjian

远程过程调用中间件(remote procedure call middleware)　基于远程过程调用技术支持分布式过程间互联互通互操作的**分布计算中间件**。

远程过程调用中间件采用同步通信方式,是过程式程序设计风范在分布式应用中的扩展。远程过程调用中间件规定了规范化的接口描述语言 IDL,服务方使用 IDL 可以方便地描述其可供客户方调用的每个过程接口。中间件则通过其 IDL 工具编译产生接口代码,生成客户方和服务方的框架函数,从而基于远程过程调用机制,便可实现跨平台的远程过程调用。

对于远程过程调用中反映网络故障的异常事件,如超时等,远程过程调用中间件将给出应答。

远程过程调用中间件结构相对简单,易于实现和使用,因此在早期规模不大的分布式系统中得到了广泛应用。

参考文献

1. Bernstein P A. Middleware: a model for distributed system services. Communications of the ACM, 1996, 39(2): 86-98

2. Tilevich E, Smaragdakis Y. NRMI: natural and efficient middleware. In: 23rd IEEE International Conference on Distributed Computing Systems (ICDCS'03), 2003: 252

(滕猛)

yuancheng wangluo jiankong

远程网络监控(remote network monitoring, RMON)　IETF 定义的标准监控规范,用来增强**简单网络管理协议**(SNMP)的功能,实现对网络进行远程监视的技术。RMON 的数据主要用于网络的性能和故障的监测,是网络规划、性能优化和协助网络错误诊断的重要数据来源。

由于 SNMP 在获取网络性能参数时,需要定期进行轮询,这样可能会产生大量的通信,给网络带来不必要的负载。RMON 则是使用各种网络监控器和管理系统之间交换的网络监控数据,网络监视器负责对网络数据进行监控、收集、分析和存储,并对网络状况不断进行测试,将采集到的信息记录反馈给网络管理站,同时 RMON 在执行和诊断功能上也对 SNMP 进行了扩展,改善了监测的性能,大大降低了对网络的开销。

RMON 系统的网络探测器(probe)分为两类,一种是专用的 RMON 探测器,网络管理端直接从探测器获取管理信息并控制网络资源,这种方式能够获取的信息较多,支持的功能全面;另一种是 RMON 代理(agent),由网络设备(如一些高端的路由器、交换机等)内置实现,收集相关的信息,该方式受设备资源限制,能够取得的数据往往有限制。

RMON 探测器能获得的信息遵照 RMON MIB。RMON MIB 由一组统计数据、分析数据和诊断数据组成,不像标准 MIB 仅提供被管对象大量的原始数据,它提供的是经过分析处理后的统计数据和计算结果。目前的 RMON 有两个版本:RMON v1 和 RMON v2。RMON v1 得到了较为广泛的支持,多数的网络设备都提供此版本的支持。它定义了 9 个 MIB 组用于基本的网络监控;RMON v2 是 RMON 的扩展,专注于物理层以上协议单元的流量,主要对 IP 层流量和应用程序层流量进行监测。RMON MIB 还有一些变种,如 SMON MIB 主要用来为交换网络提供分析。

RMON MIB 对网段数据的采集和控制通过控制表和数据表完成。RMON MIB 按功能进行分组,每个组有自己的控制表和数据表,其中控制表可读写、

数据表只读,控制表用于描述数据表所存放数据的格式。配置的时候,由管理站设置数据收集的要求,存入控制表。开始工作后,RMON 监控端根据控制表的配置,把收集到的数据存放到数据表。RMON 在监控元素的多个 RMON 组中传递信息,各个组通过提供不同的数据来满足网络监控的需要。每个组都是可选项,所以,厂商不必在 MIB 中支持所有的组。

另一方面,与 SNMP 主要关注设备本身的管理不同,RMON 着重关注网络中通过的流量相关数据。RMON 主要用于对一个网段乃至整个网络中数据流量的监视,这和 Netflow、Sflow、IPFIX 等网络流信息监测技术相似。所不同的是,RMON 能得到的信息较丰富,对硬件本身的要求也较高。

RMON 能够更好地监测网络的运行情况,同时减少 SNMP 轮询对网络带来的负载,是对 SNMP 最重要的增强,是目前应用相当广泛的网络管理标准之一。

参考文献

1. RMON1: RFC 2819—Remote network monitoring management information base

2. RMON2: RFC 2021—Remote network monitoring management information base Version 2 using SMIv2

3. SMON: RFC 2613—Remote Network monitoring mib extensions for switched networks

4. Overview: RFC 3577—Introduction to the RMON family of MIB modules

(周昌令)

yuancheng yidong kongzhi

远程移动控制(remote mobile control) 控制器和被控对象可以从物理上分离开,通过有线或无线网络相连,实现远程控制的一种控制技术。也称为**网络控制系统** NCS(networked control systems)。

网络控制系统 网络控制系统是从最早的传统**集散控制系统**发展为**现场总线控制系统**,再到嵌入式设备网络控制系统,控制系统的结构发生了变化。网络控制系统是一种开放式、分布式的反馈控制系统,传感器、控制器及执行器和被控对象通过 Internet 或无线网络实现数据传输、资源共享和协调操作。网络控制系统的一般结构如图 1 所示。

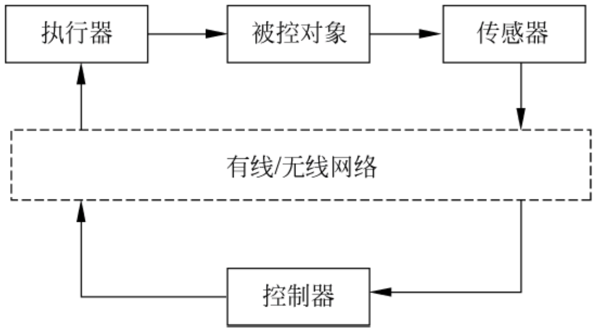


图 1 网络控制系统的一般结构

目前由**嵌入式系统**自身实现 Web 服务器功能,使得每个设备都可与 Internet 相连,是远程控制系统发展的趋势。图 2 所示的是一个远程控制系统的具体实现。由图可见,远程控制系统由多台服务器组成,服务器负责提供现场控制服务,客户机负责定时采集数据,监视服务器状态并对服务器进行远程控制。服务器采用上、下位机的结构,上位机对应于被监控的服务器主机,下位机对应于一块插入服务器 PCI 插槽中的标准 PCI 嵌入式插卡。客户机对服务器的监控是通过这块多功能控制卡完成的。客户机在远程通过 Internet 网,在浏览器中输入服务器嵌入式控制卡的 IP 地址,就可以动态地访问开发板上的动态网页,从而完成与服务器的数据通信。而控制卡通过 RS232 接口,完成与上位机的数据通信。

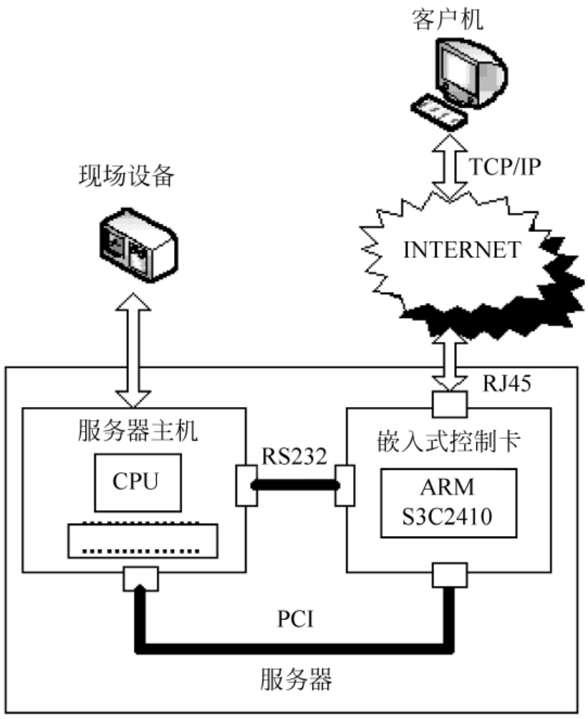


图 2 一个远程控制系统的具体实现

网络控制系统带来诸多优点的同时,也存在很多问题,如网络诱导时延、数据包丢失、时序错乱等。这些问题会导致系统性能恶化,甚至不稳定。

因此都是对网络控制系统的理论研究和实践值得重视的课题。

无线传感器网络 WSN　传感器网络系统通常包括传感器节点(sensor node)、汇聚节点(sink node)和管理节点(management node),能够通过自组织方式构成网络。大量传感器节点随机部署在监测区域内部或附近,能够实时感知监测诸如光强、温度、湿度、噪声等信息,并对这些信息进行处理,然后经过多跳后路由到汇聚节点,最后通过互联网、卫星或移动通信网络到达管理节点。用户通过管理节点对传感器网络进行配置和管理,发布监测任务以及收集监测数据。传感器节点所带的能量直接影响整个网络的健壮性和生命周期。

目前应用较广泛的短距离通信技术有:

(1) ZigBee　一种新兴的短距离、低速率无线通信网络技术。ZigBee 网络具有三种功能的节点:协调器节点(coordinator)、全功能节点(FFD)和精简功能节点(RFD),其中精简功能节点负责终端数据的采集和发送,全功能节点负责建立路由和数据的转发,协调器节点负责网络与计算机间的通信。

(2) Wi-Fi 技术　Wi-Fi(wireless fidelity,无线保真技术)是一种短程无线传输技术,能够在近 100 m 的范围内支持互联网接入的无线电信号。Wi-Fi 工作在 2.4 GHz 频段,所支持的速度最高达 54 Mb/s。

Wi-Fi 是由 AP(access point,称为网络桥接器或接入点)和无线网卡组成的无线网络。它是传统的有线局域网络与无线局域网络之间的桥梁,因此任何一台装有无线网卡的 PC 或嵌入式设备,均可通过 AP 去分享有线局域网络甚至广域网络的资源。

(3) 蓝牙　一种支持设备短距离通信(一般 10 m 内)的无线电技术。能在包括移动电话、PDA、无线耳机、笔记本电脑等众多设备之间进行信息交换。利用"蓝牙"技术,能够有效地简化移动通信终端设备之间以及设备与 Internet 之间的通信,从而数据传输变得更加迅速高效。蓝牙系统以 Ad Hoc 方式工作,每个蓝牙设备都可以在网络中实现路由选择的功能,可以实现移动自组织网络。

信息物理融合系统　信息物理融合系统(cyber-physical systems,CPS)是一类将计算机、网络与物理过程密切整合的系统,汽车电子控制系统、智能电网、智能机器人技术都属于这个范畴。从广义上可理解为这是一个在环境感知的基础上,深度融合了计算、通信和控制能力的可控可信可扩展的网络化物理设备系统。

CPS 通过人机交互接口实现计算进程和物理进程的交互,并使用网络化空间以安全的实时远程协作方式操控一个物理实体。此系统包含了将来无处不在的环境感知、嵌入式计算、网络通信和网络控制等系统工程,使物理实体具有计算、通信、精确控制、远程协作和自治功能。CPS 的核心是 3C(计算机、通信、控制)的融合。

CPS 的典型应用是汽车电子。人们对汽车功能与性能的要求体现在嵌入式物理设备 ECU(电子控制单元)上,对交通信息的获取,以及车-车,车-路交互,通过智能交通(ITS)信息系统实现,加上海量信息处理与诱导,构成了信息物理融合系统。

CPS 应用非常广泛,涉及多科学的理论与技术问题,包括理论建模和计算抽象;CPS 领域语言、程序设计自动化和高可信软件;组件、系统集成与验证;分布式传感、计算、通信和网络控制;物理接口和新型人机交互;环境感知;数据管理与挖掘;从数据到知识;安全、隐私和信任等。

参考文献

1. 岳东,等. 网络控制系统的分析与综合. 北京:科学出版社,2007

2. 刘祥志,等. 信息物理融合系统. 山东科学,2010,23(3)

（慕春棣）

yun cunchu

云存储(cloud storage)　指通过**存储虚拟化**技术、集群技术、网格技术或分布式文件系统等的功能,将网络中大量异质异构的存储设备进行统一管理、协同工作,共同对外提供数据存储和数据访问功能。它是在**云计算**(cloud computing)概念上延伸和发展而来的一种技术。

如同云状的广域网和**互联网**一样,云存储对使用者来讲,不是指某一个具体的设备,而是指一个由众多存储设备和服务器所构成的集合体。用户使用云存储,并不是使用某一个存储设备,而是使用整个云存储系统带来的一种数据访问服务。因此从用户的角度看,云存储是一种存储服务,而从服务提供者的角度看,云存储是一种新的基础架构。而为了提供云存储服务,服务提供者必须构建云存储平台。

"云存储"平台应能实现以下三方面的功能。

(1) 提供海量存储资源　需要服务提供者架设出规模巨大的全球化的存储中心,能够实现"海量"数据的存储,同时提供高安全性、高可靠性和高可用性,保障高度的隐私性。另外,它还应是高效的、低

价的、节省能源的。

（2）提供多元化数据服务　云存储除了完成传统的数据存储功能外，由于面向的是PB（千万亿字节）级甚至EB（百亿亿字节级）的海量数据，还需具有智能性，即以数据为中心，提供大规模数据存储、分享、管理、挖掘、搜索、分析等服务。

（3）提供开放的接口　云存储平台提供的接口包括两部分：面向云存储平台供应商或维护运营商，主要用于云存储平台的扩展性，即不同供应商遵循相同的接口标准即可实现平台的互联和扩充；面向用户，即用户遵循一定的接口标准，就可实现用户对云存储中存储资源的访问和使用。

与网络的层次结构类似，云存储体系结构也是分层次的。如图1所示，云存储目前一般粗略地分为四个层次：①底层是存储基础架构层，主要包括各类存储资源和相应的存储软件，以提供一个统一管理的海量存储空间。②中间层是存储服务层，包括存储空间的分配和组织管理，数据归档、数据备份、数据共享、数据加密、数据压缩、重复数据删除、信息挖掘等面向不同应用需求的存储资源组织与管理。③接口层提供安全接入和应用接口，其中安全接口包括访问控制、身份认证、权限管理和反病毒入侵等多种安全策略；应用接口包括传统的存储访问接口，如网络文件共享协议NFS和CIFS、Web服务接口，各种应用软件（如数据备份、网盘等）服务接口等。④存储访问层，主要在用户端，通过各种应用软件体现。如个人空间服务，企业空间租赁服务；数据备份、数据归档软件；企业数据、业务数据集中存储，安全防护等视频数据集中存储、数据共享服务；网址、邮件服务、IPTV等各类网络应用在线存储等。

典型的云存储服务包括数据容灾备份、网盘、空间租赁等。企业使用存储服务可节省开销，个人使用则在于提供硬件设备和物理存储空间。比如，网络管理员使用存储服务进行备份，他能够指定网络上的哪些数据需要备份和多长时间备份一次（备份频率），而不是维护一个大型**磁带库**和在公司外安排一个磁带存储保管库。用户和存储服务提供商就服务的品质、水准、性能等方面签订服务等级协议（service level agreement，SLA），根据协议，服务提供商根据每GB数据的存储开销和每份数据传输开销将存储空间租给用户。用户数据将在特定时间通过存储服务提供商的专用广域网或者因特网自动传输。如果公司数据出现损坏或者丢失，网络管理员可以联系存储服务提供商要求得到一份数据的备份。

服务等级协议（SLA）一般包括以下内容：①参与各方对所提供服务及协议有效期限的规定；②服务提供期间的时间规定，包括测试、维护和升级；③对用户数量、地点以及/或提供的相应硬件的服务的规定；④对故障报告流程的说明，包括故障升级到更高水平支持的条件，应包括对故障报告期望的响应时间的规定；⑤对变更请求流程的说明，可能包括完成例行的变更请求的期望时间；⑥对服务等级目标的规定；⑦与服务相关的收费规定；⑧用户责任的规定；⑨解决与服务相关的不同意见的流程说明。

云存储服务的指标除了要考虑存储空间、存储成本、可访问性、安全性、移动性以外，还需要考虑与数据所有权、归档、发现和搜索相关的角色和职责。

以Amazon的S3产品为例，其全称为亚马逊简

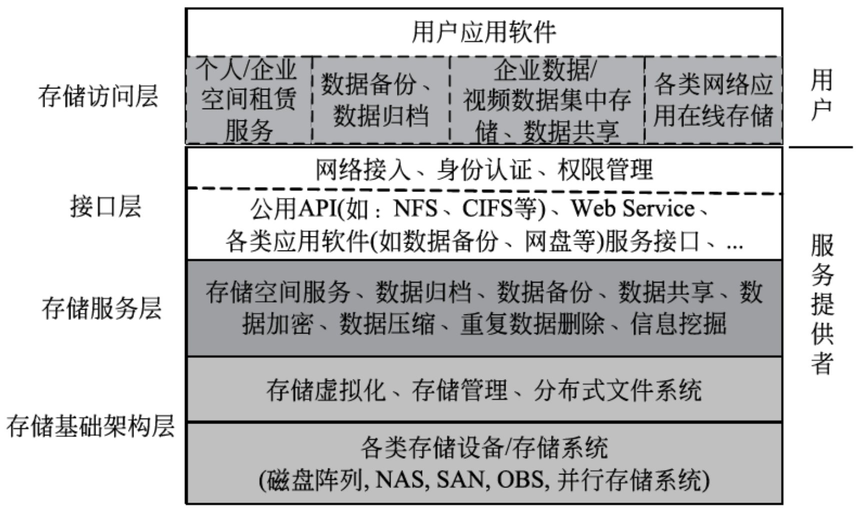


图1　云存储层次结构

易存储服务(Amazon simple storage service)。这是一个公开的服务,使Web开发人员能够轻松存储数字资产(如图片、视频、音乐和文档等),以便在应用程序中使用。经由Web服务界面,用户能够轻易把档案储存到网络服务器上。用户通过S3存储和检索的数据被称为对象,对象存储在存储段(bucket)中。若用硬盘来进行类比:对象相当于文件,而存储段相当于文件夹(或目录)。与硬盘一样,对象和存储段也可以通过统一资源标识符(uniform resource identifier,URI)来进行查找。

谷歌也有相应的云存储(cloud storage)。谷歌云存储是一项为开发人员提供存储和访问Google的云服务,也面向中小企业及大型企业提供云端数据存储空间。开发商可以通过谷歌云存储的REST API或使用任何可用的谷歌云存储工具进入可扩展的存储和网络基础设施。

参考文献

中国电子学会云计算专家委员会. 云计算白皮书, 2010

(冯丹)

yun jisuan

云计算(cloud computing)　一种商业计算模式。它将计算任务分布在由大量计算机构成的资源池上,使用户能够按需获取计算力、存储空间和信息服务。

这种资源池称为"云"。"云"是一些可以自我维护和管理的虚拟计算资源,通常是一些大型服务器集群,包括计算服务器、存储服务器和宽带资源等。云计算将计算资源集中起来,并通过专用软件实现自动管理,无须人为参与。用户可以动态申请部分资源,支持各种应用程序的运行,有利于提高效率、降低成本和技术创新。云计算的核心理念是资源池,这与早在2002年就提出的网格计算池(Computing Pool)的概念非常相似。网格计算池将计算和存储资源虚拟成为一个可以任意组合分配的集合,池的规模可以动态扩展,分配给用户的处理能力可以动态回收重用。这种模式能够大大提高资源的利用率,提升平台的服务质量。

之所以称为"云",是因为它在某些方面具有现实中云的特征:云一般都较大;云的规模可以动态伸缩,它的边界是模糊的;云在空中飘忽不定,无法也无须确定其具体位置,但它确实存在于某处。之所以称为"云",还因为云计算的鼻祖之一Amazon公司将大家曾经称为网格计算模式,取了一个新名称"弹性计算云"(Elastic Computing Cloud),并取得了商业上的成功。

有人将这种模式比喻为从单台发电机供电模式转向了电厂集中供电的模式。它意味着计算能力也可以作为一种商品进行流通,就像煤气、水和电一样,取用方便,费用低廉。最大的不同在于,它是通过**互连网络**进行传输的。

云计算是并行计算、分布式计算和**网格计算**的发展,或者说是这些计算科学概念的商业实现。云计算是虚拟化、效用计算,将基础设施作为服务IaaS(Infrastructure as a Service)、将平台作为服务PaaS(Platform as a Service)和将软件作为服务SaaS(Software as a Service)等概念结合演进并跃升的结果。

云计算的特点

(1)超大规模　"云"一般具有相当大的规模,Google云计算已经拥有100多万台服务器,Amazon、IBM、微软和Yahoo等公司的"云"均拥有几十万台服务器。"云"能赋予用户前所未有的计算能力。

(2)虚拟化　云计算支持用户在任意位置、使用各种终端获取服务。所请求的资源来自"云",而不是固定的实体。应用在"云"中某处运行,但实际上用户无须了解应用运行的具体位置,只需要一台**笔记本计算机**或一个**个人数字助理**(PDA),就可以通过网络服务来获取各种超强能力的服务。

(3)高可靠性　"云"使用了数据多复本容错、计算节点同构可互换等措施来保障服务的高可靠性,使用云计算比使用本地计算机更加可靠。

(4)通用性　云计算不针对特定的应用,在"云"的支撑下可以构造出千变万化的应用,同一片"云"可以同时支撑不同的应用运行。

(5)高可伸缩性　"云"的规模可以动态伸缩,满足应用和用户规模增长的需要。

(6)按需服务　"云"是一个庞大的资源池,用户按需购买,像自来水、电和煤气那样计费。

(7)极其廉价　"云"的特殊容错措施使得可以采用极其廉价的节点来构成云;"云"的自动化管理使数据中心管理成本大幅降低;"云"的公用性和通用性使资源的利用率大幅提升;"云"设施可以建在电力资源丰富的地区,从而大幅降低能源成本。因此"云"具有前所未有的性能价格比。

云计算与网格计算的关系是一个受关注的问题。网格是20世纪90年代中期发展起来的下一代互连网络核心技术。网格技术的开创者Ian Foster将之定义为"在动态、多机构参与的虚拟组织中协

同共享资源和求解问题”。网格是在网络基础之上,基于SOA,使用互操作、按需集成等技术手段,将分散在不同地理位置的资源虚拟成为一个有机整体,实现计算、存储、数据、软件和设备等资源的共享,从而大幅提高资源的利用率,使用户获得前所未有的计算和信息能力。网格计算与云计算各有侧重,如表1所示。

表1　网格计算与云计算的比较

	网格计算	云计算
目标	共享高性能计算力和数据资源,实现资源共享和协同工作	提供通用的计算平台和存储空间,提供各种软件服务
资源来源	不同机构	同一机构
资源类型	异构资源	同构资源
资源节点	高性能计算机	服务器/PC
虚拟化视图	虚拟组织	虚拟机
计算类型	紧耦合问题为主	松耦合问题
应用类型	科学计算为主	数据处理为主
用户类型	科学界	商业社会
付费方式	免费(政府出资)	按量计费
标准化	有统一的国际标准OGSA/WSRF	尚无标准,但已经有了开放云计算联盟OCC

云计算的服务类型

云计算按照服务类型大致可以分为3类:将基础设施作为服务IaaS、将平台作为服务PaaS和将软件作为服务SaaS,如图1所示。

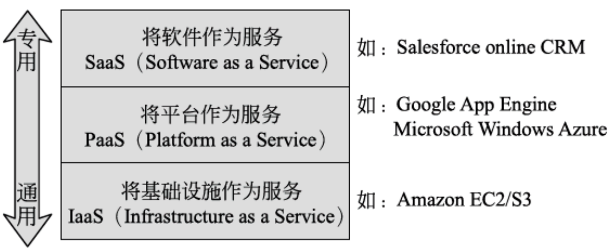


图1　云计算的服务类型

1. IaaS将硬件设备等基础资源封装成服务供用户使用,如Amazon云计算AWS(amazon web services)的弹性计算云EC2和简单存储服务S3。在IaaS环境中,用户相当于在使用裸机和磁盘,既可以让它运行Windows操作系统,也可以让它运行Linux操作系统,因而几乎可以做任何想做的事情,但用户必须考虑如何才能让多台机器协同工作起来。AWS提供了在节点之间互通消息的接口简单队列服务SQS(simple queue service)。IaaS最大的优势在于它允许用户动态申请或释放节点,按使用量计费。运行IaaS的服务器规模达到几十万台之多,用户因而可以认为能够申请的资源几乎是无限的。同时,IaaS是由公众共享的,因而具有更高的资源使用效率。

2. 将平台作为服务(PaaS)对资源的抽象层次更高,它提供用户应用程序的运行环境,典型的如Google App Engine。微软的云计算操作系统Microsoft Windows Azure也可大致归入这一类。PaaS自身负责资源的动态扩展和容错管理,用户应用程序不必过多考虑节点间的配合问题。但与此同时,用户的自主权降低,必须使用特定的编程环境并遵照特定的编程模型。这有点像在高性能集群计算机里进行MPI编程,只适用于解决某些特定的计算问题。例如,Google App Engine只允许使用**Python语言**和**Java语言**、基于称为Django的Web应用框架、调用Google App Engine SDK来开发在线应用服务。

3. 将软件作为服务(SaaS)的针对性更强,它将某些特定应用软件功能封装成服务,如Salesforce公司提供的在线客户关系管理CRM(client relationship management)服务。SaaS既不像PaaS一样提供计算或存储资源类型的服务,也不像IaaS一样提供运行用户自定义应用程序的环境,它只提供某些专门用途的服务供应用调用。

需要指出的是,随着云计算的深入发展,不同云计算解决方案之间相互渗透融合,同一种产品往往横跨两种以上类型。例如,Amazon Web Services是以IaaS发展的,但新提供的弹性MapReduce服务模仿了Google的MapReduce,简单数据库服务SimpleDB模仿了Google的Bigtable,这两者均属于PaaS的范畴,而它新提供的电子商务服务FPS和DevPay以及网站访问统计服务Alexa Web服务,则属于SaaS的范畴。

云计算的发展现状

由于云计算是多种技术结合演进的结果,其成熟度较高,又有大公司推动,发展极为迅速。Google、Amazon、IBM、微软和Yahoo等大公司是云计算的先行者。云计算领域的众多成功公司还包括VMware、Salesforce、Facebook、YouTube、MySpace等。

Amazon研发了弹性计算云EC2(Elastic Computing Cloud)和简单存储服务S3(Simple Storage Serv-

ice)为企业提供计算和存储服务。收费的服务项目包括存储空间、带宽、CPU 资源以及月租费。月租费与电话月租费类似,存储空间、带宽按容量收费,CPU 根据运算量时长收费。在诞生不到两年的时间内,Amazon 的注册用户就多达 44 万人,其中包括为数众多的企业级用户。

Google 是最大的云计算技术的使用者。Google 搜索引擎就建立在分布于 200 多个站点、超过 100 万台的服务器的支撑之上,而且这些设施的数量正在迅猛增长。Google 的一系列成功应用平台,包括 Google 地球、地图、Gmail、Docs 等也同样使用了这些基础设施。采用 Google Docs 之类的应用,用户数据会保存在互连网络上的某个位置,可以通过任何一个与互连网络相连的终端十分便利地访问和共享这些数据。目前,Google 已经允许第三方在 Google 的云计算中通过 Google App Engine 运行大型并行应用程序。Google 值得称颂的是它不保守,它早已以发表学术论文的形式公开其云计算三大法宝:GFS、MapReduce 和 Bigtable,并在美国、中国等高校开设如何进行云计算编程的课程,2010 年 4 月,Google 公开了其云计算平台监控系统 Dapper 的实现技术,紧接着在 2011 年 1 月,Google 又公开了 Megastore 分布式存储技术。相应的,模仿者应运而生,Hadoop 是其中最受关注的开源项目。

IBM 在 2007 年 11 月推出了“改变游戏规则”的“蓝云”计算平台,为客户带来即买即用的云计算平台。它包括一系列自我管理和自我修复的虚拟化云计算软件,使来自全球的应用可以访问分布式的大型服务器池,使得数据中心在类似于互连网络的环境下运行计算。IBM 正在与 17 个欧洲组织合作开展名为 RESERVOIR 的云计算项目,以“无障碍的资源和服务虚拟化”为口号,欧盟提供了 1.7 亿欧元作为部分资金。IBM 已在全球范围内建立了 13 个云计算中心,并且已帮助数个客户成功部署了云计算中心。

微软紧跟云计算步伐,于 2008 年 10 月推出了 Windows Azure 操作系统。Azure(译为“蓝天”)是继 Windows 取代 DOS 之后,微软的又一次颠覆性转型——通过在互连网络架构上打造新云计算平台,让 Windows 真正由 PC 延伸到“蓝天”上。Azure 的底层是微软全球基础服务系统,由遍布全球的第四代数据中心构成。目前,微软已经配置了 220 个集装箱式数据中心,包括 44 万台服务器。微软在 2010 年 10 月的 PDC 大会上,公布了 Windows Azure 云计算平台的未来蓝图,跳出单纯的基础架构作服务的框架,将 Windows Azure 定位为平台作服务:一套全面的开发工具、服务和管理系统。它可以让开发者们致力于开发可用和可扩展的应用程序。微软将为 Windows Azure 用户推出许多新的功能,不但能更简单地将现有的应用程序转移到云中,而且可以加强云托管应用程序的可用服务,充分体现出微软的“云” + “端”战略。

在我国,云计算发展非常迅猛。2008 年 11 月 25 日,中国电子学会专门成立了云计算专家委员会。2009 年 5 月 22 日,中国电子学会隆重举办首届中国云计算大会,此后每年举办一届,盛况空前。中国移动研究院率先建立起 1024 个 CPU 的云计算试验中心,并于 2010 年 5 月发布了“Big Cloud”1.0;阿里巴巴集团也成立了专注于云计算领域研究和研发的阿里云公司,提供类似 AWS 的云计算服务;百度云盘已经拥有数千万用户;2013 年,云创存储率先推出了单机架容量超过 2.3 PB 的超低功耗云存储系统和万亿记录量级、秒级响应的数据立方大数据库。

作为云计算技术的一个分支,云安全技术通过大量客户端的参与和大量服务器端的统计分析来识别病毒和木马,取得了巨大成功。360 安全卫士、瑞星、趋势、卡巴斯基、McAfee、Symantec、江民、Panda、金山等均推出了云安全解决方案。值得一提的是,云安全的核心思想,与早在 2003 年就提出的反垃圾邮件网格非常接近。

参考文献

1. Armbrust M, Fox A, Griffith R, et al. Above the clouds: a Berkeley view of cloud computing. mimeo. UC Berkeley, RAD Laboratory, 2009

2. Foster Ian, Kesselman C, Tuecke S. The anatomy of the grid: enabling scalable virtual organizations. International Journal of High Performance Computing Applications, 2001, 15(3): 200-222

3. Liu P, Shi Y, Li San-Li, Computing pool—a simplified and practical computational grid model. the Second International Workshop on Grid and Cooperative Computing (GCC 2003), Shanghai, Dec 7-10, 2003. LNCS 3032, Heidelberg: Springer-Verlag, 2004

4. Liu P, Shi Y, Francis C M Lau, Wang Cho-Li, Li San-Li. Grid demo proposal: AntiSpamGrid. IEEE International Conference on Cluster Computing, Hong Kong, Dec 1-4, 2003, selected as one of the ex-

cellent Grid research projects for the GridDemo session, 2003 （陈康）

yundong buhuo bianji he chongyong

运动捕获、编辑和重用（motion capture, editing and retargeting） 采用软硬件系统记录表演者的真实运动信息，经过处理后，把动作过程复制到一个虚拟的人或动物上。

与通过交互方式设置的动画不同，运动捕获（motion capture）采用软硬件系统记录表演者的真实运动信息，并把动作过程复制到一个虚拟的人或动物上。运动捕获也可看成是一种正运动学方法。该方法类似于早期 Disney 公司制作卡通片《白雪公主》时使用的 rotoscoping 技术，即动画师根据素材画面，用手工交互的方式跟踪获取画面主体的运动信息。运动捕获可以获取表演者动作的个性和细节，是生成逼真人体动画最实用、最有效的方法，因而在影视特技、游戏和动画广告中被广泛采用。该技术不仅使得人体角色动画的制作过程变得非常简单，而且可以提高角色动画的质量。在《指环王》系列电影中，古鲁姆（Gollum）便是一个虚拟的角色，其肢体和表情动作大部分通过捕获 Andy Serkis 的表演来得到。捕获的人体关节运动数据通常采用 BioVision 公司的 BVH 格式来存储。

运动捕获通常采用下面三种技术来实现。①光学运动捕获系统。将反射标记附加在表演者身体的关节处，然后用高分辨率摄像机跟踪标记的位置，从而计算出关节运动的参数。对于脸部表情数据，需要的摄像机数为 1～2 个；对于全身的关节数据，需要的摄像机数为 3～16 个或更多。光学运动捕获系统具有测量范围大、标记放置适应性强（甚至可放于大象、足球、垒球等上面）、表演者行动自由等优点，属于运动捕获系统中的高端产品。但光学运动捕获系统也具有标记容易被表演者自身遮挡、摄像机定标困难、对光线敏感、价钱贵、难以实时捕获数据等缺点。②磁性运动捕获系统。把传感器放置于表演者的身上，并测量由发射源产生的低频磁场。传感器和发射源通过电缆与一个电子控制单元相连。通常在每个人身上放置 6～11 个（或更多）传感器。磁性传感器不仅可以传回位置信息，而且可以传回旋转信息。系统采用逆运动学计算人体各个关节的角度，并补偿传感器与真实关节旋转中心的偏差。磁性运动捕获系统能实时测量运动信息，但具有对金属敏感、电缆妨碍表演者的运动、传感器易滑动、低采样率等缺点。③机械式运动捕获系统。在人的关节处放置电位计，通过电缆直接得到关节处的空间位置。

运动捕获数据是表演者真实运动的映像，因而在获取运动数据时，应尽量使表演者与虚拟角色的关节结构尺寸接近。在表演者和虚拟角色的关节结构尺寸（如身高、腿长、手臂长度等）不相同时，若不加修改地直接把运动捕获的数据直接应用到一个不同尺寸的虚拟角色上，就有可能出现运动不协调、双脚离地等不真实的运动。运动重用可以把一个角色的动画赋给另一个具有相同关节结构但具有不同关节长度的角色，并能保持原有动画的质量。

参考文献

1. 彭群生，金小刚，万华根，等. 计算机图形学应用基础. 北京：科学出版社，2009

2. http://en.wikipedia.org/wiki/Motion_capture

（耿卫东）

yundong buhuo xitong

运动捕获系统（motion capture system） 跟踪和记录人体或物体运动过程的软、硬件系统。通常使用传感器采集运动所引起的声、光、电磁场或受力变化的信息，并传输到计算机求解出被跟踪对象各个时刻的位置和姿态。最早在 20 世纪 70 年代后期用于生物力学分析，90 年代中期在影视、游戏的制作中得到广泛应用。运动捕获系统能够快速、准确地获取真实的运动数据，使计算机动画中角色的运动姿态更加逼真。运动捕获系统也是很多**虚拟现实**和**增强现实**系统的重要组成部分。

根据所采用定位技术的不同工作原理，运动捕获系统分为以下 4 种。

1. 光学运动捕获系统

使用最广泛的运动捕获系统。通常由连接到计算机的摄像机、光学标志物以及定位软件构成，利用光学定位和计算机视觉技术获取运动数据。根据光学标志物的不同，这类运动捕获系统分为运动标志物跟踪系统和固定标志物跟踪系统。

运动标志物跟踪系统包括固定在被跟踪运动对象各个部位上随对象一起运动的小型标志物，以及固定在运动空间周围支架上拍摄运动过程的多台摄像机。标志物有被动与主动两种。被动标志物一般用后向反射（retroreflective）材料制作，光源紧挨摄像机镜头照射运动对象，摄像机采集运动对象的反射光信息。主动标志物多用发光二极管制作，在对

象运动过程中依次快速地循环发光。利用同一时刻从多个不同位置拍摄的二维图像中同一标志物的位置,用三角化(triangulate)方法可以计算出该标志物的三维空间位置,从而得到被跟踪对象的位置和姿态。

固定标志物跟踪系统中,摄像机固定在人体或运动物体上,测量出运动对象与摄像机的相对位置,并把由不同特殊图案组成的光学标志物固定或投影到运动空间中,它们能被摄像机拍摄到而又不影响对象运动。利用计算机视觉技术,从摄像机拍摄的图像序列中识别出这些特殊图案,并计算出摄像机的位置和姿态,从而得到运动对象的运动信息。

2. 机械运动捕获系统

该系统由金属杆和传感器构成的局部或全身外骨架,以及相应传感器融合软件等组成,通过测量人体各个关节的角度,实时获取人体运动姿态。**传感手套**就是一种常见的局部机械运动捕获装置。在运动数据采集过程中,被采集者穿戴上局部或全身外骨架运动,位于各个关节的电位传感器将关节弯曲角度等数据传输到计算机,求解出肢体各部分的运动姿态。与光学运动捕获系统相比,具有更高的实时性,而且不需要摄像机和标志物,能捕获的运动范围仅受线缆长度或无线数据连接有效距离的限制。但是,机械外骨架不如人体关节灵活,穿戴上这类装置之后运动者往往动作变得不自然,而且各关节的活动范围也受到限制。

3. 惯性运动捕获系统

惯性运动捕获系统由微型陀螺仪、生物力学模型和传感器融合软件等组成。陀螺仪固定在运动物体或人体关节的各部位,将运动中测得的陀螺仪朝向信息通过无线方式传输到计算机上,计算出运动对象的运动姿态。与机械运动捕获系统一样,这类系统对运动范围限制少,传感器更小巧便携,对运动的影响更小。但是由于对角速度和加速度的测量误差在空间和时间上的积累,导致测量精度较低,容易出现定位飘移等现象。

4. 电磁运动捕获系统

电磁运动捕获系统由电磁传感器、定位软件组成。电磁传感器包括随物体运动的发送端和固定的接收端,它们各有三个方向相互垂直的线圈,通过记录运动过程中发送和接收端之间相对磁通量的变化,可以计算出发送端在各个时刻的位置和姿态。这类系统的优点是传感器发出的信号不会被非金属物体遮挡,传感器可以在视觉上隐藏起来。其缺点是极易受到发射变化电磁波的装置(如CRT显示器、手机等)干扰,因此一般放置在极端屏蔽的环境中。与上述各种捕获系统相比,这类系统捕获范围最小。

参考文献

1. Shiratori T, Park H S, Sigal L, et al. Motion capture from body-mounted cameras. ACM Trans on Graphics, 2011, 30(4)

2. Field M, Stirling D A, Naghdy F, Pan Z. Motion capture in robotics review. IEEE International Conference on Control and Automation, 2009, 1697-1702

3. Furniss M. Motion Capture. Dec 1999. http://web.mit.edu/comm-forum/papers/furniss.html

(齐越)

yundong fenxi

运动分析(motion analysis) 从两帧或多帧图像中抽取、估计图像序列所蕴含的运动信息的过程。在机器人导航、运动目标跟踪、行为识别和医学影像分析等领域有广阔应用前景。其主要内容包括六个方面。

运动检测 检测图像中的所有运动像素点,得到一个二值图像,一般可将运动点设为1,其他点设为0。还可做一些后处理,如噪声去除、相邻点聚类和目标标记等。对于运动检测,主要有两种方法:背景减除方法和帧差方法。

光流计算 计算图像中每个点在两帧内的相对运动。光流计算一般采用微分方法,即利用图像序列的时间导数和空间导数估计光流。基于不同的优化准则,微分方法可分为全局和局部两类方法。局部方法,如Lucas-Kanade方法,采用类似局部能量的优化函数,光流计算较精确,鲁棒性较好;全局方法,如Horn-Schunck方法,最小化全局能量函数,可产生稠密光流,但对噪声较敏感。

目标跟踪 检测特征点或目标在图像中的位置,获取运动轨迹,它需要实现目标(特征点)在连续帧中的对应。对于多目标(特征点)的对应,可使用路径一致性约束缩小搜索空间。该假设认为目标不会在图像序列中产生剧烈运动变化。著名的跟踪算法包括:卡尔曼滤波算法和粒子滤波算法。

自运动估计(Egomotion estimation) 从图像序列中,获取摄像机在场景中的三维刚体运动信息(旋转和平移)。自运动估计在自治机器人导航

领域有重要应用,如估计无人车相对于道路标志的位置等。自运动估计可首先基于光流计算伸焦点(Focus of expansion,FOE)估计摄像机三维平移运动参数,进而估计摄像机旋转运动参数。

从运动恢复结构 分析图像中的局部运动,恢复目标三维结构。通常,运动中的结构恢复需要首先进行图像中的特征点(如角点)的跟踪,之后利用特征点轨迹恢复目标的三维结构。运动中的结构恢复可为许多重要应用提供方法上的支持,如人体运动分析中,可从人体运动特征点(如关节点)在二维图像的投影,恢复三维人体结构与运动信息,进而理解人的运动行为。

参考文献

1. Sonka M, Hlavac V, Boyle R. Image processing: Analysis and machine vision. 2nd ed. CL-Engineering, 1998

2. Irani M, Rousso B, Peleg S. Recovery of ego-motion using image stabilization. IEEE Conference on Computer Vision and Pattern Recognition, 1994, 454-460

3. Dellaert F, Seitz S, Thorpe C, Thrun S. Structure from motion without correspondence. IEEE Conference on Computer Vision and Pattern Recognition, 2000, 557-564

4. Wang L, Hu W M, Tan T N. Recent developments in human motion analysis. Pattern Recognition. 2003, 36(3): 585-601

(王亮)

yundong genzong

运动跟踪(moving object tracking) 从图像序列中根据运动信息跟踪物体的过程与方法,又称运动物体跟踪或运动目标跟踪。运动跟踪通常由物体表示、动态模型、特征度量、跟踪方法四部分组成,通过推理过程完成。物体表示处理“跟踪什么”的问题,它可采用各种形状表示(如质心、矩形框、椭圆、轮廓、三维关节模型等)或表观表示(如纹理块、颜色直方图等)等来描述物体。动态模型处理“在什么范围内跟踪”的问题,可采用自回归模型(autoregressive model)等来描述物体的运动,以限定跟踪范围。特征度量处理“用什么跟踪”的问题,可采用从图像中提取的低层特征(如颜色、边缘、纹理块、光流等)与物体表示间的相似程度来给出。跟踪方法处理“怎么去跟踪”的问题,它可以分为自底向上和自顶向下两大类方法。自底向上的方法包括:①爬山法定位法,即根据动态模型把前一帧的物体信息预测到当前帧中,然后用爬山法局部搜索最佳匹配位置、表示等,其中包括基于 mean shift、snake、level set、AAM(active appearance model)等方法。②检测规约法,即先通过检测器给出当前帧中可能与物体相关的部分,然后利用关于物体的先验知识和动态模型把它们组合成有意义的物体表示。例如先检测出人体手臂、头、躯干等部分,再利用人体结构和前一帧的预测信息组装成完整人体。自顶向下的方法主要是基于状态空间模型(state space models)的方法,它包括三个部分:状态 X_t(对应于物体表示)、状态的演变模型 $P(X_t|X_{t-1})$(对应于动态模型)、观测模型$P(O_t|X_t)$(对应于特征度量),它的目标是估计已知当前所有观测 $O_{1:t}$时状态 X_t的后验概率 $P(X_t|O_{1:t})$(跟踪的输出可以是使后验概率最大的最优 X_t或基于后验概率的 X_t的期望值)。当 $P(X_t|X_{t-1})$ 和$P(O_t|X_t)$是线性高斯模型时,可采用卡尔曼滤波器(Kalman Filter)来准确估计后验概率;当 $P(X_t|X_{t-1})$ 和 $P(O_t|X_t)$是非线性高斯模型时,可采用扩展卡尔曼滤波器(extended Kalman filter)来近似估计后验概率;而当 $P(X_t|X_{t-1})$ 和 $P(O_t|X_t)$是非高斯模型时,则需采用基于离子滤波器(particle filter)或马尔可夫链蒙特卡罗(Markov chain Monte Carlo)的方法来近似估计后验概率。运动跟踪是**计算机视觉**中关键的中层处理环节,可广泛应用于视觉监控、人机交互、自动控制、运动捕获、体育运动分析、生物、军事等领域。

参考文献

1. Forsyth D A, Ponce J. 计算机视觉:一种现代方法. 林学訚, 王宏, 等译. 北京:电子工业出版社, 2004

2. Blake A, Isard M. Active contours. Springer, 1998

3. Yilmaz A, Javed O, Shah M. Object tracking: a survey. ACM Computing Surveys, 2006, 38(4), Article 13

(邱慧军)

yundong guji

运动估计(motion estimation) 估计图像序列中由于场景中物体运动造成的相邻帧间运动向量的过程(参见**图像序列处理**)。运动向量可以是关于整幅图像(即估计全局运动)、每个像素(即估计运动场)、或图像中的特定部分(即某些矩形块或特征点的运动)的。当场景与摄像机间发生相对运动

时,它在图像中的投影产生的亮度图案也随之移动,这就是*光流*。它是可看得到的亮度图案的运动。运动估计可通过光流计算或特征点对应来实现。光流计算的方法包括基于图像时空微分的方法、基于频域变换的方法、基于相关性的区域匹配方法等。特征点对应的方法通过特征点匹配或特征点跟踪来实现。特征点匹配是先从每一帧中提取一些显著特征,然后通过帧间特征的匹配来给出特征点的运动信息。特征点跟踪的方法则是在某些帧上检测特征,然后在后续帧中搜索它们的最佳对应位置(可通过梯度下降法来实现),从而给出特征点的运动信息。基于显著特征点的运动估计方法的优点是估计结果一般比光流法可靠、准确,但是缺点是只能在特征点得到度量;而基于光流的方法优点是可得到稠密的数据,但是准确性较差。运动估计的结果可以用于视频压缩中的运动补偿、**从运动恢复结构**、动作识别等。

参考文献

1. Beauchemin S S, Barron J L. The computation of optical flow. ACM Computing Surveys, 1995, 27(3): 433-466

2. Tomasi C, Kanade T. Detection and tracking of point features. Carnegie Mellon University Technical Report, CMU-CS-91-132, April 1991

3. Zitova B, Flusser J. Image registration methods: a survey. Image and Vision Computing, 2003, 21: 977-1000

(邱慧军)

yundong jiance

运动检测(motion detection)　把场景中有变化的部分(称为前景)与其他部分(称为背景)分离开的过程和方法(参见**图像序列处理**)。运动检测的输入是对同一个场景在不同时间拍摄的图像序列,目的是要检测出每帧图像中的前景,以便进一步对其进行后续处理。运动检测可广泛用于视觉监控、远程遥感、医疗诊断、水环境监测、驾驶辅助系统等领域。如视觉监控中,运动检测可以从摄像机连续拍摄的图像中把运动物体检测出来,以便进一步分析其行为;医疗诊断中,运动检测可以从人眼视网膜的间隔几个月的不同成像中检测出视网膜组织的萎缩等。运动检测是序列图像处理中的一个重要的基本步骤,通过前景和背景的分离,可有效简化后续相关处理。

在对同一个场景不同时间拍摄的图像序列中,导致图像变化的因素有:①场景内容的变化,如物体的出现与消失,物体相对于背景的运动,物体形状的变化等;②成像条件的变化,如摄像机运动、光照变化、传感器误差等。运动检测的目标是要检测出有意义的变化,即主要关心场景内容的变化,而不是成像条件的变化。为了把成像条件变化导致的图像变化过滤掉,在运动检测过程中需要考虑相关的预处理。如需要考虑图像对准的问题来消除摄像机运动的影响,或采用光照归一化、滤波、光照建模等手段来应付光照变化的因素。需要指出的是,如果运动检测方法能够容忍相关的变化,则不需要相应的预处理。

运动检测通常可通过两种方法来实现。一种方法是简单差分法:直接把当前帧图像与前一帧图像或背景图像相减,如果某个像素的图像差值大于一定阈值,则检测为前景像素。这种方法比较形象直观,但是对于噪声和光照变化等比较敏感。另一种方法是假设检验法:如通过判断每个像素属于背景的概率是否小于某个阈值来给出前景检测,或通过判断每个像素属于前景的概率与属于背景的概率的比值是否超过一定阈值来给出前景检测等。由于在相关概率建模时可以综合考虑各种因素(如可通过混合模型来描述光照变化等导致的图像变化),假设检验法的性能优于简单差分法。此外,由于运动检测一般是在每个像素上独立进行的,检测结果很难保证空间一致性,如存在孤立点、洞、锯齿边界等。这时,可通过中值滤波、形态学操作等后处理,或在检测方法中考虑空间关系的先验信息(如基于马尔可夫随机场的方法)改善检测结果。

参考文献

Radke R J, Andra S, Al-Kofahi O, Roysam B. Image change detection algorithms: a systematic survey. IEEE Trans on Image Processing, 2005, 3(14): 294-307

(邱慧军)

yundong tuxiang de yasuo bianma biaozhun

运动图像的压缩编码标准(compression and coding standards for motion image)　对随时间变化的图像序列(又称动态图像)进行压缩编码的技术标准,也称为视频编码标准。动态图像实时地记录了对象的动态变化过程,需要每秒25帧~30帧图像来表示,因此动态图像的数据量十分巨大。但是在序列中帧与帧之间存在高度的相关性,变化往往发生在局部空间内。如果能够对运动变化部分

用运动向量来描述，那么某一帧的图像就可以看成它的前帧图像经过运动向量补偿后的结果。因此，运动估计与补偿是动态图像压缩编码的主要手段。

运动图像压缩编码分为帧内压缩和帧间压缩两部分。帧内压缩大多是基于离散余弦变换（DCT）的静态图像压缩技术（参见**图像的压缩编码**），减少空域冗余度。帧间压缩把图像序列分为帧内图（I）、前向预测图（P）、双向预测图（B）三种图像。帧内图以静态图像压缩方法处理，是基础图像。前向预测图用前面的帧内图根据运动矢量进行预测补偿，因此主要传送预测的差值。双向预测图可以根据前面和后面图的信息进行前向、后向或者双向预测。可以看到，仅有帧内图和运动矢量需要传送，其余的可由插补和补偿来完成，有相当大的压缩率。由于实际系统中运动图像压缩算法的计算量大，开发专用硬件系统（芯片）是动态图像压缩技术的一个重要方向。

运动图像压缩编码的国际标准主要有两个系列：国际标准化组织（ISO）建议的 MPEG 标准和国际电信联盟 ITU 的 H.26x 标准。

H.120：这是第一个视频编码国际标准，由 ITU-T 在 1984 年正式通过，它的应用码率范围为 1544～2048 kb/s。主要采用的编码方法为：条件填充（conditional replenishment），标量量化（SQ），可变长度编码（VLC）。1988 年又通过了第二版。

H.261：这是第一个被广泛使用的视频编码国际标准，由 ITU-T 在 1991 年正式通过，它的码率为 80～320 kb/s。主要采用的编码方法为：16×16 宏块运动补偿，8×8 块的 DCT，标量量化（SQ），二维 Run-Level 可变长度熵编码（RLVLEC）。1993 年又通过了第二版，目标码率为 64～2 048 kb/s。

MPEG-1 Video：一个成功的并被广泛使用的视频编码标准（IS-11172-2），质量和 VHS 相当或者稍好，码率为 1.5 Mb/s。它是由 ISO/IEC JTC1 MPEG 运动图像专家组在 1993 年制定的。在编码技术上引进了双向运动预测编码（B 帧），半像素精度运动矢量。在高码率时性能比 H.261 要好。

MPEG-2 Video/H.262：一个成功的并被广泛使用的视频编码标准（IS-13818-2），和 MPEG-1 相比码率和图像质量都要高得多，支持标准电视和高级电视的分辨率，码率为 4～30 Mb/s。它是由 ISO/IEC JTC1 MPEG 运动图像专家组和 ITU-T 在 1994 年制定的。它包括 MPEG-1 标准并提供了隔行扫描图像的有效压缩方法，主要是场模式运动估计补偿方法，另外还有分层的码率扩展功能。和 MPEG-1 不同的是 MPEG-2 系统层（IS-13818-1）将码流的传输分为节目流和传输流两种。其中节目流用于误码率低的应用环境中，它只包括一个节目；传输流可用于误码率较高传输环境（传输介质），可以同时包括多个节目。

H.263：这是第一个针对超低码率的编码标准，最初预定码率是 10～30 kb/s，在制定过程中扩展到 10～2 048 kb/s，ITU-T 在 1996 年通过了 H.263 的 1.0 版本，H.263 技术上的新特点是可变块大小的运动补偿，覆盖的块运动补偿（OBMC），图像外插运动矢量，三维的 RUN-LEVEL-LAST 可变长度编码（RLVLC），中值运动矢量预测，更有效的头信息等。在低码率时（低于 30 kb/s），它只需一半码率便可获得与 H.261 相同的图像质量。

H.263 有 2 个增强版本：H.263+（H.263 2.0 版）和 H.263++（H.263 3.0 版）。H.263+ 对无线和基于包（packet-based）传输网络提供高度容错保护，它是第一个具有这项功能的编码标准，此外在编码效率、灵活的视频输入格式和可扩展性方面都有所增强。H.263++ 又增加了三项可选的功能：①一种增强的参考帧选择模式，它可以提高压缩效率和容错性能；②一种数据分配切片模式，它是一种数据组织方法，能够提高容错性能；③定义了一些附加的增强信息，它可以保持向后兼容。

MPEG-4 Part 2：一个针对交互式多媒体通信的视频编码的国际标准（IS-14496-2），它包括了所有以前标准的编码方法（包括静态图像编码方法），另外新增加了静态纹理的小波编码、图像分割后的形状编码、合成视频和自然视频的混合编码等。它包括了各种码率、图像分辨率、帧率并支持隔行扫描格式。低码率时它的效率和 H.263 类似并与之兼容，而高码率时和 MPEG-2 相当，这是第一个定义交互的通信标准。支持各个视频对象同时编码，在解码器端支持视频编辑等功能。MPEG-4 还针对无线网络传输增加了容错和扩展编码的功能，这些都是新技术。目前无线网络传输已经可以使用 MPEG-4 标准，另外它还增加了许多复杂的运动预测技术如全局运动补偿预测技术、2D Mesh 编码等。

MPEG-4 AVC/H.264：一个专门针对网络多媒体通信传输的国际标准，由 ISO/IEC MPEG 和 ITU-T 共同制定，2003 年正式通过（"improved video coding for multimedia communication"）。该标准的目标是在各种码率条件下比 MPEG-2，H.263，或 MPEG-4

Part 2 等标准的压缩比提高一倍。它采用多种不同大小块的运动补偿和 1/4 像素精度的运动向量,并针对低码率下块效应采用了环内低频滤波,以减少块效应的影响。

自 2003 年标准制定以来,MPEG-4 AVC/H.264 不断进行修改和完善。例如为适应网络环境变化的需求,2007 年增加了“可分层视频编码 SVC”(附件 G),它对视频进行分级编码,通过在传输基本层码流的基础上,选择性地传输增强层码流,从而动态适应网络带宽。为了对不同视点拍摄同一景物的视频信号进行编码,提供用户立体感和交互体验,2009 年 MPEG-4 AVC/H.264 标准增加了多视点视频编码 MVC(附件 H),它通过引入视点间的参考,进一步提高了编码的效率。

参考文献

1. 姚庆栋,毕厚杰,王兆华,等. 图像编码基础. 北京:人民邮电出版社,1994

2. 钟玉琢,王琪,贺玉文. 基于对象的多媒体数据压缩编码国际标准——MPEG-4 及其校验模型. 北京:科学出版社,2000　　(孙立峰　钟玉琢)

yunsuanqi

运算器(arithmetic unit)　计算机的数据处理中心,也是机器中各部件交换数据的枢纽。现代计算机的奠基人冯·诺依曼在世界上第一台电子计算机 ENIAC 研制过程中,于 1945 年发表了《电子计算机逻辑结构初探》的报告,提出了建造现代计算的结构模型,要点是:计算机由五大部件即运算器、**控制器**、存储器、输入装置和输出装置组成,运算器为中心;采用二进制编码表示数据;提出了指令的二进制表示方法和存储程序的概念等,奠定了现代计算机的理论基础,提出的模型被称为冯·诺依曼计算机结构模型。

运算器的地位

冯·诺依曼强调五大部件运算器为中心有两层意义:其一,计算机主要用途是计算,运算器是专门用于计算的部件,是计算机的核心,其他部件都是为运算器服务的。其二,运算器是机器中各部件交换数据的枢纽,输入数据、输出数据的指令都是通过运算器实现的。处理数据时是从存储器中读出数据,又将运算结果写回到存储器中。**辅助存储器**磁盘和**主存储器**交换数据可以利用直接存储器存取(DMA)控制器实现直接存储器访问,这是为了不破坏运算器现场提高运算速度而增加的替代部件,当磁盘向**中央处理器**(CPU)发出 DMA 请求,CPU 响应后放弃系统总线控制权转入 DMA 周期,在 DMA 控制下占用一个**机器周期**交换一个数据后,又把总线交给 CPU。继续执行 CPU 原来的工作。

运算器的组成

运算器的核心是算术逻辑部件(ALU),其本质上是一个二进制并行加法器。减法也可在加法器中实现,乘除法可通过加减法和移位完成。为了提高运算速度,在加法器输出端增加一级移位器,使加法和移位在一个时钟周期内完成。为了完成逻辑运算,在加法器的数据输入端增加与门、或门等实现与、或等逻辑操作。在程序中有时需要使用前条指令的运算结果,如结果为 0、结果溢出、结果有进位等,作为分支程序条件转移的依据,因此 ALU 还设置了一个程序状态字存储器(psw)专门存放运算结果的特征,运算器中还设置若干数据寄存器,存放被运算数和运算结果。完成加减法操作最少需要二个寄存器,完成乘除法运算最少需要 3 个寄存器,这些寄存器都具有专门的职能。显然增加寄存器的数目,对方便编程和提高运算速度是有利的。现代计算机中往往设置多个不固定功能的通用寄存器,每个时钟周期同时读出二个操作数,可提高速度。运算器内各部件的数据传送通过总线方式连接,构成完整的数据传送通路,这组总线称为处理器内总线,也叫处理器总线。运算器基本逻辑框图如图 1 所示。

为了提高运算速度,运算器中还可增加乘法部件,除法部件,浮点部件等。

运算器分类

运算器有多种分类方式。按处理一个字内各位数据的顺序,运算器可分为串行运算器和并行运算器。按数据寄存器的功能可分为基于累加器的运算器和基于通用寄存器方案的运算器。累加器用来存放运算结果(也可放一个被运算数据),指令中不再指明,常常用于一地址指令中。按被处理数据类型可分为定点运算器和浮点运算器。按处理器内总线的组数,可分为单总线、双总线、三总线方案,以进一步提高运算速度,三总线方案可使 ALU 两个数据输入端及输出端放置不同数据,在一个节拍中完成双操作数的运算。

现代计算机中采用流水线概念研制成流水线处理器。其原理是按指令操作过程分成若干流水级,设置专门的装置完成指定流水级的操作,构成流水线处理部件,理想情况下每一时钟周期,可得到一条

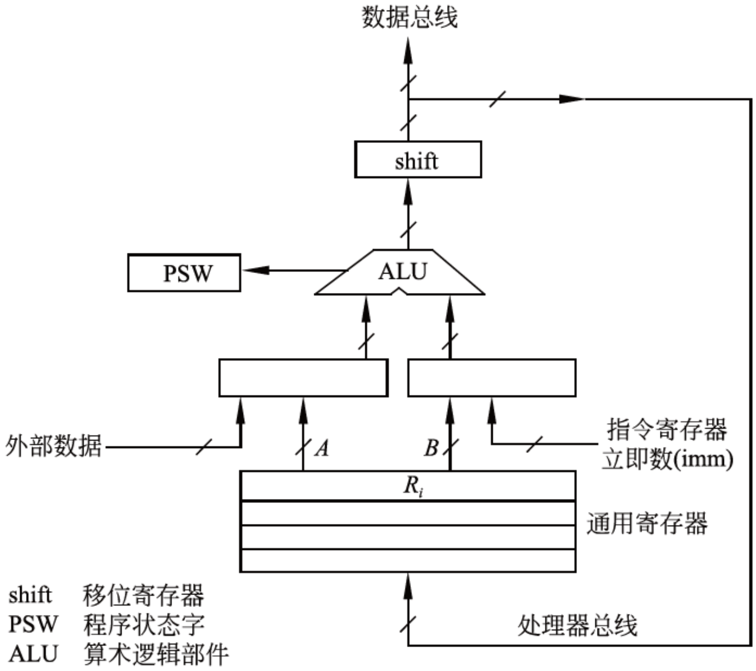


图 1 运算器基本逻辑框图

指令处理结果。在超级流水线结构中,每一个时钟周期可以处理多条指令,运算器结构更为复杂。在超标量超流水线结构中处理器中可设置多个处理单元多条流水线。

参考文献

1. 金兰,金波. 计算机组织:原理、分析与设计. 北京:清华大学出版社,2006

2. 王爱英. 计算机组成与结构. 4 版. 北京:清华大学出版社,2007

(谢树煜)

yunsuan sudu pingjia

运算速度评价(arithmetic speed evaluation)

采用计算分析、模拟、测试等方法和工具,对计算机系统执行算术逻辑运算的速度进行推测和测定。其中,计算机系统运算速度是系统工作能力和生产效率的主要表征,是设计者和使用者共同关注的重要性能指标,通常用每秒执行指令的条数来表示。

早期的计算机系统以算术运算为主要操作,常用加法指令的执行速度来表征机器运算速度。但是随着计算机系统日趋复杂,仅用加法指令已难以评估机器的运算速度。1959 年,吉布森提出从应用课题程序中统计各类指令所占百分比,并用指令混合比计算指令的平均执行速度,称吉布森混合法。对不同应用领域,程序类型不同导致所用指令的混合比也不相同,其比值分别根据不同领域的大量应用程序统计归纳得出。人们常采用 MIPS[百万(条)指令/每秒]和 MFLOPS 值[百万(次)浮点运算/每秒]来表征计算机的平均运算速度,即让计算机执行按一定比例混合的各种基准程序,从而计算出整数平均运算速度和浮点平均运算速度(参见**系统性能指标**)。由于系统运算速度与用户实际观察到的差距很大,于是人们模拟用户的实际负载并抽出常用算法,进而编制通用测试程序,用以测试不同类型、不同型号的计算机系统的运算速度。

运算速度评价方法很多,包括:①计算各种指令执行速度。根据单处理器主时钟频率 f(MHz),求出单处理器基本工作节拍 T,$T=\frac{1}{f}$(ns)。再根据处理器结构模型和指令操作流程,推算出执行各种指令的基本节拍和每秒执行指令的次数。②计算吉布森混合比运算速度。假定第 i 类指令($i=1,2,\cdots,n$)在使用过程中出现的概率为 P_i,其执行时间为 t_i,则平均执行指令时间 t_E 为 $t_E=\sum_{i=1}^{n}P_it_i$。t_E 的倒数即吉布森混合比平均运算速度,其中 P_i 的集合即为混合比。③计算峰值运算速度。假设指令和数据都在高速缓存内,全部流水线无阻塞、全部功能部件都满载工作的最佳环境下,计算出处理器理论设计的最高运算速度。④计算核心程序运算速度。经过实际统计,为数不多的指令占程序总运行时间

90% 以上这种程序称核心程序。核心程序的运算速度更能反映机器在该应用中实际的运算速度。⑤计算典型程序运算速度。选取实际应用中有代表性的课题,如 FFT、图形图像处理等,计算出典型程序运算速度。

运用模型分析和模拟方法,可以计算较为近似的系统平均运算速度。模型分析方法为计算机建立一种用数学方程表示的模型,进而在给定不同参数的条件下,分析和计算影响系统运算速度的因素。计算机系统由一组有限的资源组成,可以用排队论来描述资源争用的现象,并用概率论计算共享资源对运算速度的影响。模型模拟方法通过构造模型来逼近目标系统,包括系统模型和工作负载(环境)模型,它们是相互联系和相互影响的,一般采用面向对象技术和程序描述语言来描述。模拟模型建立后,需要验证它的合理性、正确性,还需要设计模拟试验,对结果进行分析。

使用示波器、逻辑分析仪和测试程序等硬、软件工具,对计算机系统进行实际测试,是最基本、最直接、最重要的系统运算速度评价手段。主要包括:硬件监视,在处理器设置(或外接)指令监视器记录执行指令条数和运行时间;软件监视,用相应测试程序监视执行指令条数和运行时间;参考机比较,选定一个已知的、公认计算速度的系统做参考机,再选定一组测试程序进行比较。

参考文献

1. Hennessy John L, Patterson David A. 计算机体系结构:量化研究方法(原文第 5 版). 贾洪峰,译. 北京:机械工业出版社,2012

2. 李三立,李亚民. RISC——单发射与多发射体系结构. 北京:清华大学出版社,1993

3. 罗晓沛,侯炳辉. 系统分析员教程. 北京:清华大学出版社,1992 (陈玉霜)

yunxingshi yanzheng

运行时验证(runtime verification) 一种监测与分析计算系统(软件和硬件)执行行为的轻量级形式化技术。该技术通过提取计算系统的运行信息,监测并分析系统行为是否满足给定的性质;在一些情况下,还包括制导系统做出适当的反应,以避免性质违反时产生的系统失效。

运行时验证在形式化方法框架下通过监测程序的执行行为进行动态分析。它使用形式化规约表达性质,既可以是逻辑规约(如时序逻辑),也可以是操作规约(如有穷状态机)。对于可检测性质的形式化逻辑规约,运行时验证基于形式语言与自动机理论进行算法综合,可自动生成相应的监测器,并通过插装技术将其注入到待监测系统中。由于运行时验证是分析系统动态执行中的行为,故而避免了模型检验和定理证明等形式验证技术的系统建模复杂性,也回避了遍历全部系统行为的状态爆炸问题,具有良好的可扩展性;另一方面,与传统测试技术相比,由于采用形式化技术,其在监测系统行为时具有更好的系统性和较高的自动化程度。同样,由于属于动态分析方法,运行时验证也存在系统行为难以全覆盖、特定性质是否可监测等固有问题。

运行时验证技术在安全策略的监测、系统故障的保护与恢复等领域已得到较为广泛的应用。近来,在大规模和复杂系统的开发中,随着体系结构设计中监控结构的运用,运行时验证技术正与设计综合技术融合,成为系统设计技术的重要组成部分。总体上看,运行时验证技术仍处于发展阶段,亟须研究解决的问题包括减少运行时验证的开销,增强可验证性质的表达能力,扩展对系统行为的运行时预测分析能力等。

参考文献

Runtime verification. In: Wikipedia, http://en.wikipedia.org/wiki/Runtime_verification (王戟)

yunzuo guocheng

运作过程(operation process) 用户和操作人员在用户的业务运作环境中为了使系统或软件产品投入运行所进行的一系列有关的活动。此过程包括对系统或软件产品的运作活动和用户运作时对他的支持活动。此过程的目的是在软件开发过程完成后,将该系统从开发的环境移到用户的业务运作环境中运行;在运行时对用户的要求提供帮助和咨询;并对运作效果作出评价。此过程可为开发过程和维护过程提供有关的反馈信息,作为改进系统的依据。运作管理者可根据软件项目的总体要求按照软件**管理过程**的内容对运作过程进行管理。

运作过程一般包括以下活动:实施过程的准备;运作测试;系统转移到业务运作环境中;系统运作;对用户运作的支持;系统运作评价;用户运作评价等。

(1) 在开始准备时,要了解清楚软件开发过程的结果、准备用户的业务运作环境、制定运作过程的实施计划和运作评价标准。实施计划包括任务的分

配、过渡计划(例如考虑人机并行运行)、操作培训计划、运作步骤等。

(2) 为了使系统转移到用户的业务运作环境中运行,操作者必须在业务运作环境中对系统进行运行测试。如果其测试结果能满足运行的基准要求,则可交付系统。

(3) 在系统向业务运作环境转移的过程中,必须进行以下工作:编制有关转移的文档;进行数据的转移;进行程序的转移;进行试运行;在试运行过程中进行跟踪;进行业务转移。当这些工作都完成后,进行转移的确认,并将转移结果的评价通知开发人员。

(4) 系统投入正式运行后,必须进行管理。运作管理者和操作者应当收集运行的数据,判别、记录和解决运作中发现的问题,并改进运作环境。

(5) 操作者应当给予用户以运作上的支持。这包括:对用户进行操作培训和其他培训;应当建立接收、记录和解决用户请求的步骤;对用户的请求提供帮助和咨询服务,并对这些请求的响应进行记录和监控;必要时,操作者应当根据用户的请求向软件**维护过程**提供改进信息或修改请求,并由软件维护过程进行解决。

(6) 系统运作评价主要是指对系统的质量指标和其他方面评价。例如系统的响应分析、系统的运行效率、系统的安全性和保密性、系统故障分析、数据及媒体的管理以及人员管理、系统管理、运作时间管理等。

(7) 用户运作评价包括用户所要求的业务功能的实现程度、使用系统的容易程度、用户所设置的资源的运行和管理、系统运作效果以及用户对系统的改进要求等。

参考文献

1. IEEE Standard for Developing Software Life Cycle Processes — IEEE Std. 1074—1991. 1991

2. ISO /IEC 12207:1995 Information Technology— Software — Part 1: Software Life-Cycle Process. 1995

(刘光龙)

Z

zaoxing jishu

造型技术(modeling technique) 采用计算机表示景物形状、结构、组织、外观和行为等的方法与技术。它是**计算机图形学**的重要内容之一,也是**计算机辅助设计**(CAD)的核心技术之一。造型技术基本包含以下几方面的内容。

几何造型是造型技术最基本的内容。几何造型通过对规整形体各部分的几何形状、空间位置以及各部分之间的连接关系进行描述建立其几何形状模型,并且通过对简单模型进行几何变换及操作,实现复杂几何模型的构建。几何造型总体上可以分为线框造型、曲面造型和实体造型三类。其中,曲面造型主要包括自由曲面、**隐式曲面**、**网格曲面**和点云曲面等造型方法,实体造型可以分为传统实体造型和参数化特征造型两种。

异质实体造型是传统实体造型的拓展。传统实体造型的对象是由均质材料构成的物体,因此并不适用于对由均质材料、复合材料、功能梯度材料和界观细结构构成的复杂非均质零件进行数字化建模。随着异质零件在航空航天等领域的应用越来越多,基于快速原型技术的异质零件制造成为可能,人们提出异质实体建模方法。异质实体建模的关键是对异质实体的几何信息和异向材质信息进行融合建模。其代表性的方法有:以边界表示为基础的丛空间方法,基于体网格模型的局部合成控制方法,基于梯度源的异质实体建模方法等。

基于物理的造型是几何造型技术的发展。在几何造型中,物体的形状和位置直接由物体的几何数据和拓扑结构来表示。而在复杂的动画制作等应用中,物体的形状和位置是运动或变化的,而且为了逼真,运动或变化应该符合物理规律。这时,靠人工定义物体的几何数据和拓扑关系非常繁杂,有时甚至无法实现。这样,就出现了基于物理的造型技术。在这一技术中,物体的运动或变化规律由物理定律描述,根据物理定律产生的运动或变化动态生成物体的几何模型。有效的运动和变化控制策略、快速稳定的物理模型数值计算方法等是实现基于物理的造型的关键。

自然景物造型是一类特别的造型技术。在计算机艺术、**计算机动画**中往往涉及大量的自然景物,例如山峦、树木、草丛、云、雾等,而这些自然景物因其固有的不规则性,难以用面向规则形体的几何造型方法进行几何建模。为此,人们提出了基于过程式生成的自然景物造型方法。即用一个简单的模型及少量的易于控制的参数来表示一大类物体,通过不断改变参数,递归调用这一模型,逐步产生出数据量很大的不规则物体,因而这一技术又称为数据放大技术。基于这一原理的自然景物造型技术有分数维造型、基于文法的造型及粒子系统等。

造型技术已经在**计算机辅助设计**(CAD)、**计算机辅助工程**(CAE)、**计算机辅助制造**(CAM)、**计算机图形学**、**虚拟现实**、**可视化**、**计算机动画**、**地理信息系统**、**计算机视觉**、**计算机仿真**等诸多学科领域获得了广泛应用,同时也在随着应用需求的提高而不断发展。主要发展趋势包括:对景物建模的广度和深度不断提高、支持对超大规模场景和高度复杂产品的建模、快捷方便的造型技术等。

参考文献

1. Mortenson M E. Geometric Modeling 2nd ed. New York: John Wiley & Sons, 1997

2. Foley J D, van Dam A, et al. Computer Graphics: Principles and Practice. 2nd ed. Addison-Wesley, 1995

3. 孙家广,等. 计算机图形学. 3 版. 北京:清华大学出版社,1998

(高曙明 杨长贵)

zengqiang xianshi

增强现实(augmented reality, AR) 将计算机生成的虚拟环境或对象的信息直接叠加在用户可感知的现实环境之上的技术。增强现实系统通常表现为现实对象为主、虚拟对象为辅的人机交互系统,用户可以通过虚拟对象来增强对真实世界的理解。这正是"增强现实"名称的来由。增强现实的"增强"特征首先体现在往所感知的真实世界中添加(叠加)虚拟对象和删除(隐蔽)真实世界中的对象。技术上,添加虚拟对象比删除真实对象较易实现。

因此当前增强现实仅指往所感知的真实世界上添加虚拟对象。其次,“增强”特征应适用于所有视、听、触(力)、嗅等感觉通道的刺激信号。

构造增强现实系统的关键在于将虚拟对象无缝地融合到真实世界中。目前有两类技术可以实现上述目标,分别是光学技术和视频技术。应用光学技术实现融合的基本原理是通过一片半透明、半反射镜片实现融合,用户眼睛通过半透明镜片看到真实世界的同时,可以看到半反射镜片上的虚拟对象。应用视频技术实现融合的原理是将虚拟对象或环境的信息叠加在摄像机采集得到的真实世界视频图像上,再统一显示出来。

增强现实的关键技术包括:①聚焦与对比度匹配　增强现实中的虚拟对象由计算机生成,所以它们不论远近总是清晰的;而摄像机摄取的真实世界图像可能部分清晰,部分模糊。另一方面,真实世界的动态对比度极大,而摄像机和图像监视器的动态对比度范围有限,因此在聚焦和对比度上都存在匹配困难问题。②系统的可移动性　增强现实的用户通常位于任务现场,且要求能在较大范围内移动,从而要求整个增强现实系统能随身携带,方便移动。③时空一致性问题　增强现实系统对配准精度的要求很高,且这种配准大多是动态的,因此对实时性要求也很高。在时间和空间上的配准误差严重时将使用户出现位置不适感。④传感器问题　为实现精确注册,要求能实时地、精确地、远程地跟踪用户头部位置和朝向,目前的传感器技术离上述要求尚有较大差距。⑤光照一致性问题　在增强现实系统中,虚拟物体绘制后的光照效果应与拍摄视频中的实景相一致,否则将会严重削弱真实感和**沉浸感**。光照一致性的实质是虚拟物体与视频中实景共享同一个或近似的光照环境。需要解决以下几个问题:实景的光照环境获取、全局光照明条件下的虚拟图形绘制、虚实景之间的阴影、摄像机感光系统的 gamma 曲线获取与应用。其中虚实景之间的阴影计算是比较难的问题,阴影投射需要求出光源、虚拟物体、实景三者之间的空间关系,这需要恢复出视频中实景的深度信息。众所周知,从视频中恢复深度是一个比较困难的问题,而且在增强现实系统中,这种深度恢复在许多场合需要在实时完成。

增强现实的应用原理在于通过将虚拟对象叠加在真实对象之上提示用户注意这些真实对象,或补充新的实际并不存在的虚拟对象,或通过相应解释性文字等信息,帮助用户理解真实对象。增强现实系统已在医学**可视化**、维护与修理指导、解释与提示、机器人路径规划、导游、娱乐、建筑工程规划、飞机导航和攻击瞄准等领域获得应用。

参考文献

1. Azuma R T. A survey of augmented reality. Presence: Teleoperators and Virtual Environments, 1997, 6(4): 355-385

2. Bimber O, Raskar R, Inami M. Spatial augmented reality. SIGGRAPH 2007 Course 17

（鲍虎军）

zengqiang xuni

增强虚拟(augmented virtuality, AV)　将现实对象(实际景物或从现实环境获取的信息或其重建模型)以适当方式叠加融合到计算机生成的虚拟环境中的一种技术。增强虚拟系统通常表现为虚拟对象为主、现实对象为辅的人机交互系统,用户可以通过现实对象来增强对虚拟世界的感知逼真性。增强虚拟的“增强”特征首先体现在往所感知的虚拟世界中添加(叠加)现实对象。技术上,这种叠加操作既可以离线进行,也可以在线进行。当然,前者比较容易,此时的增强虚拟系统与传统虚拟现实系统非常类似,但用户对增强虚拟系统的感知更具真实性;后者则相对困难,增强虚拟系统与现实环境实时同步地不断演化,用户可以在增强虚拟系统中对现实对象进行交互分析,并反馈到现实环境中。

构造增强虚拟系统的关键在于将现实对象无缝地叠加融合到虚拟环境中。其技术难点主要有三方面:①为了让虚拟对象以及现实对象在虚拟世界中的映像看起来更真,需要对现实环境信息进行高保真的采集。随着传感器技术的发展,现实环境的数据采集水平有了很大提高,采集信息(包含了几何、光照、材质及其动态演化信息等)的精度、维度和效率也越来越高,某些信息甚至可以实时采集得到,为增强虚拟技术的发展奠定了基础。类似于增强现实技术,数字摄像机也是增强虚拟系统经常采用的信息获取传感器。②为了让虚实对象能够相互融合,需要对采集信息的处理和实物信息模型的重建。由于数据采集的不完整性和庞大的数据量,如何自动完整地建立起实物的信息模型,高保真地刻画现实环境是其中的核心。在此基础上,需要高效率地建立这些信息模型,其终极目标是实现实时重构。例如,可以利用摄像机网络同步实时地从多个角度采集实际场景,进而借助三维视觉和图形技术,重建出其三维信息模型。③为了让增强虚拟环境可信,需

要对虚实对象模型进行实时融合和逼真再现。类似于**增强现实**技术,当实物直接被叠加融合到虚拟环境中时,虚实对象的时空一致性仍是问题的关键;另一困难问题是如何逼真、实时地将实物信息模型和虚拟对象模型绘制呈现出来,使得用户感觉它们与现实世界完全一样。近年来,这些技术均得到了一定的发展,但离上述要求尚有较大差距。

由于将现实世界的各种信息实时地传输融入到虚拟环境中,增强虚拟技术一方面可有效提高虚拟环境的感知逼真度,另一方面使得用户能够借助虚拟环境去在线分析和操控现实环境的演化。因此,增强虚拟技术自然扩展了传统**虚拟现实**技术在交互模拟分析方面的应用范畴,并开始在复杂系统(如远程无人装置和大型装备或生产流水线等)的模拟和在线优化操控等方面获得了应用。

参考文献

1. Azuma R T. A survey of augmented reality, Presence: Teleoperators and Virtual Environments. 1997, 6(4): 355-385

2. Kanade T, Rander P, Narayanan P. Virtualized reality: constructing virtual worlds from real scenes. IEEE Multi-Media Magazine, 1997, 4(1): 34-47

(鲍虎军)

zhan zidongji

栈自动机(stack automaton)　一种受限型的图灵机(图1)。它含有:①有限控制器;②具有端点标志的输入带和双向输入读头,输入带只读不写;③一存储带或栈,它可以用作下推存储,读头不能印或抹(即印空白),除非它的右面全是空单元。但读头可以按只读(不写)方式在带的非空部分内到处游动。

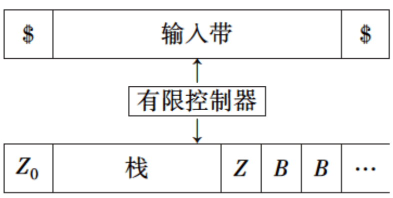


图 1　栈自动机

栈自动机类似于**下推自动机**。不同之处在于它可以在输入带上左右移动读得输入符号,也可以在栈的任何位置上读得符号,但改写符号只能在栈顶。栈自动机比下推自动机识别语言的能力强,但仍大大弱于图灵机。栈自动机一般可看成为一种非确定型的受限图灵机,可将其修改为确定型非抹除的(即任何栈符号都不被抹除)。已经证明:如果语言 L 可被一确定的非抹除的栈自动机接受,则它可被一确定的带复杂度为 $n \log n$ 的图灵机接受。栈自动机及其许多变种在接受语言方面的性质已被广泛研究。

(李祥)

zhangliangchang keshihua

张量场可视化(tensor field visualization)
将张量场蕴含的各种现象和结果以可视、直观的图形方式展现出来以揭示张量场运动变化规律的可视化方法。由坐标系改变时满足一定坐标转化关系的有序数组成的集合称为张量。张量场广泛用于数学、物理和工程领域,如微分几何、基础物理、光学、固体机械学、流体动力学、环境工程、航空航天和弥散张量成像等。张量场可视化涉及张量场的可视表示、可视呈现、特征跟踪等。

有关张量场的数学理论可用于刻画大尺度张量场的分析和可视化:①张量计算,如张量的导数、张量的分解和张量之间的距离;②张量场不变量的计算,如张量形状、轨迹、各向异性分数值、张量场模式等;③张量场结构计算,如张量场分割,张量场的峰、谷和裂缝,以及基于动力系统理论的拉格朗日连贯结构计算等。

张量场可视化方法可分为基于纹理、几何、拓扑三类。基于纹理的方法将张量场转换为一张动态演化的图像(纹理),图释张量场的全局属性。其思路是将张量场简化为向量场,进而采用线积分法、噪声纹理法等方法显示。基于几何的方法显式地生成刻画某类张量场属性的几何表达。其中,图标法将张量单个地表达为某个几何表达,如椭球和超二次曲面。超流线法将张量转换为向量(如二阶对称张量的主特征方向),再沿主特征方向进行积分,形成流线、流面或流体。基于拓扑的方法计算张量场的拓扑特征(如关键点、奇点、灭点、分叉点和退化线等),依此将感兴趣区域剖分为具有相同属性的子区域,并建立对应的图结构,实现拓扑简化、拓扑跟踪和拓扑显示。基于拓扑的方法可有效生成多变量场的定性结构,快速构造全局流场结构,特别适合于数值模拟或实验模拟生成的大尺度数据。

张量场可视化的典型案例是弥散加权核磁共振成像和流体力学计算。例如,弥散张量核磁共振成像是唯一能无创地观察活体组织的水分子通道的成像方法。它用一个对称的二阶张量表达每个成像点的弥散方向,并在此基础上计算出有意义的特征,如

各向异性分数值等。比弥散张量成像更为复杂的是张量阶数大于 2 的高角分辨率弥散张量成像。

将数据组织为张量可为分析和可视化三维空间数据提供有效的工具。例如,将一个多维向量表达为一个低阶张量,可采用张量场理论分析大尺度的数据集。将线性或多线性的分解模型应用于高阶数据张量,可有效实现数据压缩和自动分析。高阶张量也可被用来表示实值和对称函数。

参考文献

1. Hansen C D, Johnson C R. The visualization handbook. Elsevier, 2005

2. Laidlaw D, Weickert J. Visualization and processing of tensor fields: advances and perspectives. Springer, 2009
(陈为)

zhangwen shibie

掌纹识别(palmprint recongition) 计算机通过分析人手掌上的纹路信息进行自动身份识别的过程和技术。手掌(palm)是指手腕到手指根之间的掌心区域,掌纹则是手掌皮肤上所有纹路的统称,主要包括主线(principal lines)、皱褶线(wrinkles)和乳突纹(ridges)等。主线(也称屈肌线)是手掌上最粗最深的线,主要是在手掌的抓握过程中,随着手掌、手指的张开、闭合而形成的,多数人的手掌上有三条主线,分别称为生命线(life line)、智慧线(head line)和感情线(heart line)。皱褶线是手掌上比主线细、浅的不规则的纹线,有些皱褶线是天生的,而另外一些是由于长时间手的抓握而引起的肌肉运动所导致的。乳突纹是人体胚胎发育过程中随着手掌最外层表皮永久变厚形成的,遍布整个手掌,通常是平行结构,或为曲线,或为直线。掌纹具有非常高的稳定性和独特性,可有效地用于身份识别。掌纹识别具有精度高、速度快、价格低、用户接受度高等优点。

掌纹识别的研究是从 20 世纪 90 年代末由香港理工大学张大鹏教授(David Zhang)首先开始研究的,并于 2003 年研制出了第一套自动掌纹识别系统。

掌纹识别主要包括掌纹图像采集、预处理、特征提取和匹配几个部分。

在掌纹图像采集部分,主要解决高质量掌纹图像的采集问题,要求所采集的图像清晰度高,受环境变化影响小,采集速度足够快,同时要求掌纹采集方便、易于为用户所接受。目前最常用的掌纹采集系统是由香港理工大学和哈尔滨工业大学联合研制的基于电荷耦合元件(charge-coupled device,CCD)图像传感器的联机掌纹采集系统。

预处理模块主要解决掌纹图像的刚性变形问题,消除掌纹的旋转、平移和尺度变化等,最常见的是通过手掌轮廓上的某些关键点,如位于两手指根部之间的点,来对掌纹进行预处理。

特征提取和匹配部分主要解决掌纹特征的定义、提取和匹配问题,要求所提取的特征有较强的区分能力、较少的存储量、提取和匹配特征所需时间较少。这是掌纹识别研究的核心内容,目前主要包括三大类方法:①基于图像变换的方法。该类方法是将掌纹图像进行某种变换,使掌纹在变换域中可提取区分性更好的特征。②基于掌纹结构特征的方法。通过分析手掌上由主线、皱褶线或者乳突纹所形成的结构信息来实现身份识别。③基于掌纹纹理特征的方法。将整个掌纹当成纹理图像来分析,进而提取和匹配纹理特征。目前纹理特征是区分能力最强的掌纹特征之一。

掌纹识别技术虽然已经达到了实用化程度,但是多数系统要求用户将手放在距成像元件一定距离的支架上,以获取高质量的掌纹图像,从而获得高的识别精度。这使得掌纹识别系统的体积较大,其应用范围受到一定的局限。如何减小掌纹识别系统的体积,同时保持或进一步提高识别精度,是掌纹识别进一步研究和关注的内容。

参考文献

1. Zhang D. Palmprint authentication. Kluwer Academic Publishers, 2004

2. 邬向前, 张大鹏, 王宽全. 掌纹识别技术. 北京: 科学出版社,2006
(邬向前)

zhenlie chuliji

阵列处理机(array processor) 在同一控制器控制下,按同步方式工作的处理器阵列。阵列处理机是一种典型的单指令流多数据流(SIMD)计算机(参见**并行处理系统**)。

在阵列处理机中,指令由阵列控制器译码,阵列中所有的处理器都执行同样的操作,只是处理的数据不同。

1958 年,F. H. Unger 为求解空间问题设想了由一个主控制器控制的二维处理单元阵列结构;1962 年,D. L. Slotnick 等提出的 Solomon 计算机,进一步肯定了 SIMD 概念。第一台大型的 SIMD 阵列处理机是 60 年代末由伊利诺依大学设计、1972 年由

Burroughs 公司生产的 Illiac Ⅳ(64 个处理单元)。此后,Burroughs 公司又设计了性能更高的 BSP 阵列处理机。为了满足图像处理等领域的应用需求,70 年代末和 80 年代初出现了一些由大量位片处理器构成的阵列处理机系统,如英国的 CLIP 和 DAP,美国的 MPP 等。1985 年,Thinking Machine 公司的 CM-1 使用了 65 536 个位片处理单元。进入 90 年代后,MIMD 型 MPP 系统开始与 SIMD 阵列处理机竞争市场,但仍有一些阵列处理机产品问世,如 MasPar MP-1(16 384 个处理单元)和 MP-2(65 536 个处理单元)。另外,一些半导体厂家也开始推出适合图像或数字信号处理等特定应用的单片阵列处理器。

阵列处理机通常不是一台独立的计算机,它需要一台宿主机作为用户界面,并将程序和数据装入阵列处理机的控制器中的存储器。但新一代的阵列处理机,如 MP-1 等,已将宿主机功能集成在系统中。

在阵列处理机中,除了阵列控制器和处理单元阵列外,还有一个标量处理器。阵列处理机的指令一般分为标量指令和向量指令两类,阵列控制器完成指令译码后,将标量指令交给标量处理器执行,而将向量指令广播给所有处理单元执行。另外,通过条件指令可以设置屏蔽,以临时禁止某些处理单元参加运算。

阵列处理机有分布式存储器和共享存储器两种类型,前者是主流。分布存储的阵列处理机中,处理单元包含运算部件和局部存储器两部分,各处理单元间通过点对点的数据交换网络(如二维网格、超立方体等)相连。共享存储阵列处理机的典型代表是 BSP,其处理单元与存储器模块通过一个对准网络相连。为了减少存储体的冲突,BSP 使用了 17 个存储模块(即素数个存储模块)。

参考文献

黄铠,Briggs F A. 计算机结构与并行处理. 金兰,等译. 北京:科学出版社,1990　　(*唐志敏*)

zhenshigan tuxing shengcheng

真实感图形生成(realistic image synthesis, photo-realistic graphics generation)　使三维空间的物体生成具有色彩、纹理、阴影、层次等真实感图形的过程和技术,又称真实感图像综合。其目的是对于空间的各种物体和自然景物,利用计算机图形生成技术产生恰如拍照片一样的真实感效果。

为了产生图形的真实感,一般需要解决以下几方面的图形综合技术问题:

(1) 在图形中消除在特定观察点看不见的物体或部分物体,从而产生空间物体的层次感(参见**消隐技术**);

(2) 利用小的纹理样本在物体表面或三维物体中合成产生自然光滑的纹理(参见**纹理合成**);

(3) 在物体表面生成各种各样的纹理,以增强物体的质感(参见**纹理映射**);

(4) 尽可能精确地模拟光源照射的物理效果,使空间物体具有像拍照相片一样的光照效果和明暗层次(参见**光照模型**、**光线跟踪技术**、**辐射度技术**、**光子映射**方法);

(5) 产生景深与运动模糊的效果(参见**景深与运动模糊**);

(6) 产生光照阴影的效果(参见**阴影生成**);

(7) 在显示设备有限的离散精度范围内,尽量保持图形具有自然的光影过渡和连续性(参见**图形反走样技术**)。

使计算机生成的图形具有逼真的感觉是许多领域所追求的目标。该技术在**计算机动画**(如影视、广告、计算机艺术等)、**计算机仿真**、**计算机辅助设计**、**计算机辅助制造**、科学计算**可视化**以及**虚拟现实**等众多领域已得到广泛应用。真实感图形生成技术可模拟生成更为复杂的自然场景和具有更高的逼真度,特别是在影视、虚拟现实等领域得到了进一步的发展和应用。

参考文献

1. 唐荣锡,汪嘉业,彭群生,汪国昭,等. 计算机图形学教程. 修订版. 北京:科学出版社,2000

2. 孙家广,等. 计算机图形学. 3 版. 北京:清华大学出版社,1998

3. 唐泽圣,周嘉玉,李新友. 计算机图形学基础. 北京:清华大学出版社,1995　　(*吴恩华*)

zhengjiao pinfen fuyong

正交频分复用(orthogonal frequency division multiplexing, OFDM)　频分复用技术的一个扩展。它不同于传统的频分复用技术,是一种高效的多载波调制技术。其基本原理是:将发送的数据流分散到多个子载波上,使各子载波的信号速率降低,从而提高抗多径衰落的能力。各子载波的频谱有 1/2 重叠,但保持相互正交,这正是它得名正交频分的原因;在接收端,采用解调技术分离出各子载波,同时消除码间的干扰。

OFDM 技术不是新技术，20 世纪 60 年代已经有了初步的概念；70 年代，美国的一个专利实现了 OFDM 的原理，但最初仅用于军事通信系统中，且因为实现复杂，实用的脚步停止不前，直到快速傅里叶变换引入系统中，一些其他实现的难题也得到解决，OFDM 技术才重新绽放它的光彩。

OFDM 系统中，信号的处理和收发是关键，图 1 是收发原理示意图。

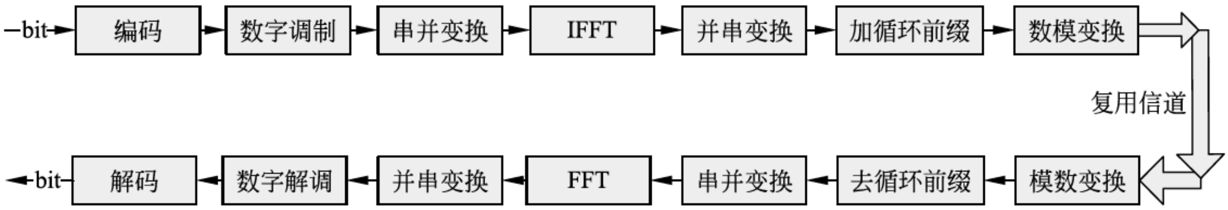


图 1　OFDM 收发原理示意图

在发送方，首先对待发送的二进制数字信号进行编码和调制，然后进行串并转换，经逆傅里叶变换后再并串转换，之后加循环前缀，即构成待发送的 OFDM 码元，再经数模转换，通常还要经过一个低通滤波器，即可输出传输的信号。

在接收方，处理的流程和发送方相反，首先经过模数转换，去循环前缀，然后进行串并变换，经过傅里叶变换后再并串转换，最后经数字解调和解码，还原出原始信号来。

需要注意的是：

（1）OFDM 不是一种新技术，它的原理早就存在，只是随着技术的进步，它的实现已不再成为难题，OFDM 重新回到人们的视野，并在高速数字通信系统中得到广泛的应用，如高速数字环路 HDSL、非对称、数字环路、高清电视 HDTV 的地面广播系统等；它也是第三代、第四代移动通信系统准备采用的技术之一。

（2）OFDM 系统具有抗脉冲干扰的能力，比单载波系统强很多，这是因为 OFDM 信号的解调是在一个较长的符号周期内积分，从而使脉冲噪声的影响得以分散。

（3）具有抗多径传播和衰落的能力，OFDM 系统将信息分散到多个载波上，降低了各载波的信号速率，使符号周期比多径迟延长，从而减少多径传播的影响，再加上通常采用保护间隔、时域均衡等措施，可以有效降低码间干扰。

（4）在 OFDM 系统中，符号周期、载波间距和子载波数应该根据实际应用条件合理选择，符号周期的大小影响载波间距及编码调制迟延。通常符号周期应该使信道在一个符号周期内稳定，子载波数应该根据信道带宽、数据速率和符号周期共同来确定。

参考文献

1. Tanenbaum A S, Wetherall D J. 计算机网络. 北京：清华大学出版社，2012

2. 纪越峰. 现代通信技术. 北京：北京邮电大学出版社，2010

3. 张辉，曹丽娜. 现代通信原理与技术. 西安：西安电子科技大学出版社，2008

（袁华）

zhengquexing weihu

正确性维护（correctness maintenance）　指为了识别、隔离和纠正计算机设备或软件系统的错误、改正性能缺陷以及排除实施中的误使用所进行的诊断和改正错误的过程。又称改正性维护或纠错性维护，目标是使失效的系统恢复到业务或服务所要求的运行状态和性能水平。

在硬件或软件系统交付使用后，通常会存在一部分潜藏的错误被遗留到运行阶段。这些潜藏的错误在某些特定的使用环境、运行条件或从未测试过的输入数据组合下会暴露出来。为此，正确性维护可用于改正在系统开发阶段已经发生，但是在系统测试阶段尚未发现的错误。正确性维护可以进一步细分为：即时正确性维护，即在错误发生后立即进行维护工作；延迟正确性维护，指维护工作根据一个给定的维护规则的集合被推迟进行。

在系统交付使用的初期，正确性维护的工作量相对较大。根据软件工程理论，正确性维护的工作量一般占据整体维护工作量的 20% 左右。一方面，正确性维护工作所发现的某些错误可能并不关键，不会影响系统的正常运行，因而其相应的维护工作可随时进行；另一方面，某些错误却至关重要，甚至会影响到整个系统的正常运行，因而其相应的维护工作必须按照规范的维护规则来制定计划，进而纠

错、复查和控制。尽管错误发现率在正确性维护工作中会不断下降，但在长期的系统使用过程中，计算机新技术的涌现以及用户需求的不断提高会加重适应性维护和完善性维护的工作量。在这些维护过程中，又可能引入新的错误，从而加重维护的总体工作量。

根据系统开发是否采用了结构化分析与设计方法，正确性维护的方法和技术有所不同。第一，对于未采用结构化分析与设计方法的系统，维护手段主要是测试和修改程序源代码，维护工作较大。第二，对于采用了结构化方法开发的系统，其交付时已有完整的软件配置文档、维护系统接口以及测试说明书等。因此，通过分析系统修改可能带来的影响，可以设计改正错误的途径。经过改正后的相应源程序还需使用测试说明书中包含的测试方案进行回归测试。同时，需要对相应的文档进行更新，避免造成改正后的源程序与文档不一致，从而影响后续的系统应用和维护工作。此外，一些新兴的自动维护技术和基于计算机网络的远程维护技术逐渐成为发展趋势。例如，利用远程维护软件自动收集系统运行信息，并使用数据挖掘技术分析程序运行时的故障信息及软、硬件相关信息，从而提供精确的纠错指导，以提高系统维护效率并降低维护成本。

参考文献

1. IEEE Standard Computer Dictionary, 610. 12, 1990

2. Department of Defense Standard Practice, MIL-STD-3034: Reliability-Centered Maintenance (RCM) Process, 2011

3. Swanson E B. The Dimensions of Maintenance. Proceedings of the 2nd International Conference on Software Engineering, San Francisco, 1976: 492-497

（刘方明）

zhengze biaodashi

正则表达式（regular expression） 对有限自动机所接受的语言或时序开关电路的行为的形式描述。它告诉我们，如何使用正则运算从原子语言构造一般语言。原子语言为空语言 φ 和单元集 $\{a\}$，其中 a 是预先指定的字母。正则运算为并、连接和连接闭包。“并”是通常集合论中的“并”运算。两个语言 X,Y 之间的连接 XY 由形如 xy 的字组成，其中 $x\in X,y\in Y$。语言 X 的连接闭包 X^* 由空字和所有形如 $x_1\cdots x_n$ 的字组成，其中 $n\geqslant 1,x_i\in X$。例如，以任意方式连接 ab 和 b 得正则表达式 $(ab\cup b)^*$，所得语言 X 为字母表 $\{a,b\}$ 上由空字和以 b 结尾且不含子字 aa 的所有字组成的语言。

正则表达式的形式定义：假设 V 和 $V_1=\{\varphi,\cup,*,(,)\}$ 是不相交的字母表。字母表 $V\cup V_1$ 上的字 α 称为 V 上的一正则表达式，恰当 α 是下述情形之一：

（1）α 是 V 的字母，或字母 φ，

（2）α 形如 $(\beta\cup\gamma),(\beta\gamma),\beta^*$ 之一，其中 β,γ 是 V 上正则表达式。

按下述约定，V 上每个正则表达式 α 表示 V 上的一语言 $|\alpha|$：

（1）φ 表示的语言是空语言，

（2）$a\in V$ 表示的语言由字 a 组成，

（3）对 V 上正则表达式 β,γ，$|(\beta\cup\gamma)|=|\beta|\cup|\gamma|$，$|(\beta\gamma)|=|\beta||\gamma|$，$|\beta^*|=|\beta|^*$。

看上去很不相同的正则表达式可标记相同语言。例如，如下正则表达式

$(a\cup ab\cup ba)^*$，$(ba\cup a^*ab)^*a^*$，$a^*(ab\cup ba^*a)^*$ 表示相同语言。

有限自动机和时序开关电路所对应的正则表达式化简后，相应行为变得很好理解。在正则表达式与有限自动机之间，存在许多相互转化的算法。

作为正则表达式的应用实例，许多操作系统、屏幕编辑器、字处理程序已扩展了它们的字符串搜寻能力，使得除寻找特殊串外，还可用来匹配一特定正则表达式的串。这种搜寻工具的最好例子是 Unix 的 grep 命令，它表示“获得正则表达式”。

参考文献

Hopcroft J E, Ullman J D. 自动机理论、语言和计算导引. 徐美瑞，译. 北京：科学出版社，1986

（陈火旺　贲可荣）

zhengze wenfa

正则文法（regular grammar） 左线性或右线性文法。

一个左线性文法可用四元组 $G=(V,\Sigma,P,S)$ 表示，其中 V 是变元的有限集合，Σ 是终结符的有限集合，$S\in V$，称为开始符号，$w\in\Sigma^*$（即 w 为有限个终结符连接成的串或字，可能为空串或空字 ε）。A，$B\in V$ 时，P 是由形为 $A\to w$ 和 $A\to wB$（$A\to Bw$）产生式组成的有限集。右线性文法与左线性文法是等价的，即可生成同样的语言（字集合）类。

正则文法来源于 20 世纪 50 年代中期N. Chomsky

对自然语言的研究,是乔姆斯基短语结构文法分层里的 3 型文法。正则文法类是上下文无关(2 型)文法类的真子类,已应用于计算机程序语言编译器的设计、词法分析(文本处理中描述触发过程动作的文本模式、文件类型和扫描器、文本工具的标准基础)、开关电路设计、句法模式识别等,是计算机和信息科学、工程、物理、化学、生物、医学、应用数学不可忽视的论题。

正则文法的结构与复杂性测度由变元、产生式的个数及文法有向图的高度、每一层的结点数来确定。$S \overset{*}{\underset{G}{\vdash}} w$ 表示有限次使用 P 中产生式可派生出字 w,正则文法 G 可作为生成器产生和描述正则语言 $L(G) = \left\{ w \in \Sigma^* \,\middle|\, S \overset{*}{\underset{G}{\vdash}} w \right\}$。例 1. $G = (\{S,A,B\},\{0,1\}P,S)$,$P = \{S\to 0A|0, A\to 1B, B\to 0A|0\}$,$G$ 是一个正则(右线性)文法,$L(G)$ 中含字 0,$(S\to 0)$,01010 $(S\to 0A\to 01B\to 010A\to 0101B\to 01010)$。正则语言也称为正则集,可以用正则表达式表示。对任一正则表达式,可以构造出带 ε 动作的非确定有限自动机(NFA)在线性时间内来接受它,也可构造出不带 ε 动作的确定有限自动机(DFA)在平方时间内来接受它,正则文法生成的语言也可由双向确定有限自动机(2DFA)来接受,NFA, DFA, 2DFA 是等价的,即所接受的语言类是相同的。

正则表达式 递归地定义如后,设 Σ 为有限集,①$\varnothing$,ε 和 $a(\forall a\in\Sigma)$是 Σ 上的正则表达式,它们分别表示空集、空字集$\{\varepsilon\}$和集合$\{a\}$;②若 α 和 β 是 Σ 上的正则表达式,则 $\alpha\cup\beta$,$\alpha\cdot\beta=\alpha\beta$ 和 α^* 也是 Σ 上的正则表达式,它们分别表示字集$\{\alpha\}$,$\{\beta\}$,$\{\alpha\}\cup\{\beta\}$,$\{\alpha\}\{\beta\}$和$\{\alpha\}^*$,运算符$\cup$,· 和 * 分别表示并、连接和星(乘幂闭包$\{\alpha\}^* = \left\{\bigcup_{i=0}^{\infty}\alpha^i\right\}$),优先顺序为 *,·,$\cup$;③只有有限次使用①和②确定的表达式才是 Σ 上的正则表达式,只有 Σ 上的正则表达式所表示的字集才是 Σ 上的正则集。如例 1 中正则文法生成的字集,其正则表达式为 $0(10)^*$。为了简化正则表达式,常用下列等式:①$\alpha\cup\alpha=\alpha$(幂等律);②$\alpha\cup\beta=\beta\cup\alpha$(交换律);③$(\alpha\cup\beta)\cup\gamma=\alpha\cup(\beta\cup\gamma)$(结合律);④$\alpha\cup\varnothing=\alpha$,$\alpha\varnothing=\varnothing\alpha=\varnothing$,$\alpha\varepsilon=\varepsilon\alpha=\alpha$(零一律);⑤$(\alpha\beta)\gamma=\alpha(\beta\gamma)$(结合律);⑥$(\alpha\cup\beta)\gamma=\alpha\gamma\cup\beta\gamma$(分配律);⑦$\varepsilon\cup\alpha^*=\alpha^*$;⑧$(\varepsilon\cup\alpha)^*=\alpha^*$。用①至④可将 α 变为 β 时,称 α 与 β 相似。正则表达式的导式类似于微积分学中的导式,可用于正则语言的商运算,α 关于 $x\in\Sigma^*$ 的导式 $D_x\,\alpha$ 递归定义如下:①$D_\varepsilon\,\alpha=\alpha$;②$\forall x\in\Sigma, D_x\varnothing=\varnothing, D_x\,\varepsilon=\varnothing, D_x\,a=\begin{cases}\varnothing, & x\neq a\\ \varepsilon, & x=a\end{cases}$ $(\forall a\in\Sigma)$;③$\forall x\in\Sigma, D_x(\alpha\cup\beta)=D_x\,\alpha\cup D_x\,\beta$,$D_x(\alpha\beta)=(D_x\alpha)\beta\cup\alpha(D_x\beta)$,$D_x(\alpha^*)=(D_x\,\alpha)\alpha^*$;④$\forall x=x_1\cdots x_n\in\Sigma^*, D_x\alpha=D_{xn}(\cdots(D_{x2}(D_{x1}\alpha))\cdots)$。每一正则表达式只有有限个不相似的导式。表 1 给出了正则表达式复杂性的两种测度。运用正则表达式方程 $X_i=\alpha_{i_0}+\alpha_{i_1}X_1+\cdots+\alpha_{i_n}X_n$ 来处理语言有其方便之处,因为这种方程中 $\Delta=\{X_1,\cdots,X_n\}$(未知量的集合)与 Σ 之交为$\varnothing$,α_{ij}为 Σ 上正则表达式,当 α_{ij}为$\varnothing$,ε 时分别相当于普通线性方程组之系数 0,1,可以按线性方程组的高斯消去法求解,当然,这里的解是一个集合,即解不是唯一的,但该算法能够正确地确定一个极小不动点作解。

表 1

测度 \ 正则表达式	$\varnothing$	ε	a	$\alpha\cup\beta$	$\alpha\beta$	α^*
H(* 在表达式中嵌套的层数——星高)	0	0	0	$\max\{H(\alpha),H(\beta)\}$	$\max\{H(\alpha),H(\beta)\}$	$H(\alpha)+1$
N(表达式中出现的符号个数)	0	0	1	$N(\alpha)+N(\beta)-\{$重复出现于 α,β 中符号个数$\}$	$N(\alpha)+N(\beta)-\{$重复出现于 α,β 中符号个数$\}$	$N(\alpha)$

正则文法的性质 由所生成的正则语言来体现。如果 R 为正则语言,则存在一个常数 n,使得 R 中所有字长不小于 n 的字 w 都可写成 xyz 的形式($y\neq\varepsilon$ 且$|xy|\leqslant n$)并且对所有非负整数 i 必有 $xy^iz\in R$,此为泵引理。它是证明某些语言不正则的有力工具,且有助于建立算法来判断一个给定的正则文法所生成的语言是有限的还是无限的。判断某些语言是否正则还可以利用对语言运算是否封闭来决定。已知正则语言类对布尔运算(并、交、补)、连接、*(克林尼闭包)、左右商、替换、同态、逆同态、

INIT(求前缀)、FIN(求后缀)、MIN、MAX、CYCLE、Reversal 等封闭。又当 $p(x)$ 为非负整系数多项式,R 为正则语言时,$L_1=\{w|$对某个使$|y|=p(|w|)$的 y 有 $wy\in R\}$,$L_2=\{w|$对某个使$|y|=|w|$的 y 有 $wy\in R\}$ 也是正则语言。当 R,R_1 和 R_2 是正则语言时下列问题都是可判定的:$w\in R$? $R=\varnothing$? $R=\Sigma^*$? $R_1=R_2$? $R_1\subseteq R_2$? $R_1\cap R_2=\varnothing$?

正则文法的推断 模式识别中研究模式类不可缺少的工具之一。即根据具体问题的特性,从其样本集 $R^+(\subseteq L(G))$ 及其否定 R^-(R^+ 关于 $L(G)$ 的补集),用算法来找出正则文法 G 就是正则文法的推断。确定的 G 不仅能生成 R^+,而且可以生成有关模式更多的样本、事实和问题的其他解答。正则文法是最容易实现推断算法公式化的文法,而且必定可以最小化。正则文法的推断方法有三:①交互作用,即借助教师(操作员)来判定字的合法性及其性质;②推广规范文法 $G_1(L(G_1)=R^+)$ 为 G;③从接受 R^+ 的有限自动机来得到 G。

并行正则语言 适应大规模并行计算的需要。仍使用语言运算来研究,除∪,·和∗三种基本运算以外,还引入了四种运算:"$\|$"(交织),$\forall a\in\Sigma$,$a\|\varepsilon=\varepsilon\|a=\{a\}$,$\forall a,b\in\Sigma,s,t\in\Sigma^*$,$as\|bt=a(s\|bt)\cup(as\|t)b$,$\forall A,B\subseteq\Sigma^*$,$A\|B=\{x|$有 $s\in A,t\in B$,使 $x\in s\|t\}$(注:含 $\|$ 的表达式可以归约为不含 $\|$ 的正则表达式);"[ ]"(同步合成),$\forall A\subseteq\Sigma_A^*,B\subseteq\Sigma_B^*$,$A[\,]B=\{x|x/\Sigma_A\in A,x/\Sigma_B\in B\}$,($x/\Sigma$ 表示字 x 在 Σ 里的符号串);"σ"(再命名),$\sigma(R)=\{\sigma(x)|x\in R\}$,如果 $\sigma(\varepsilon)=\varepsilon$ 表示 σ 可隐蔽,如果 $\sigma(a)=\sigma(b)$ 表示 σ 可以部分观察,如果 $\sigma(a)=a$表示 σ 为相似过程或遗传;"α"(Alpha 闭包),$A^\alpha=\bigcup_{i=0}^{\infty}A^{(i)}$。在上述七种运算下封闭的正则表达式与佩特里网语言相联系,从而用于研究并行计算。

***ω* 正则文法** 正则文法产生式中允许 ω 为无限串则生成 ω 正则语言,也得到不少研究成果,如豪斯朵夫-库拉托夫斯基分层、莫勒自动机分层。

正则文法与抽象代数 正则集与半群(服从结合律的二元运算的非空集)、幺半群(有幺元的半群)、态射和直积等抽象代数内容相联系已形成学科性课题,已经证明:①$R\subseteq\Sigma^*$,Σ^* 的右同余(等价)关系 E_R 规定为:$\forall x,y\in\Sigma^*$,$x\,E_R\,y\Leftrightarrow\forall z\in\Sigma^*$ 有 xz 和 yz 都在 R 中或都不在 R 中。R 为正则语言 $\Leftrightarrow\Sigma^*$ 关于 E_R 的等价类类数有限。②Σ^* 对连接运算构成一个幺半群,R 为正则语言⇔存在有限幺半群 N 和同态映射 $\varphi:\Sigma^*\to N$ 及 $T\subseteq N$ 使得 $R=T\varphi^{-1}$。③令 $PF(Q)$ 为 Q 到 Q 的全体偏函数集,$PF(Q)$ 对函数合成运算构成一个幺半群,同态映射 $\varphi:\Sigma^*\to PF(Q)$ $(x=x_1\cdots x_n\in\Sigma^*,\varphi(x)=\varphi_x\in PF(Q))$ 下 R 的象 M_R 称为 R 的语法幺半群。当 Q 为接受语言 R 的自动机 M 的状态集时,$\varphi_x(q)=\varphi_{xn}(\cdots(\varphi_{x_1}(q))\cdots)$,$q\in Q$,$\varphi_{x_i}(q)=\delta(q,x_i)$,$\delta$ 是 M 的转移函数。R 为正则集⇔M_R 为有限集。④每一正则语言 R 都存在同态映射 h_1,h_2,h_3,h_4 使 $R=h_4h_3^{-1}h_2h_1^{-1}(1^*0)$。⑤塞缪尔·爱伦堡定理:正则语言的星流形(指若干个正则语言形成的族在布尔运算、派生、逆同态下封闭)与有限幺半群流形(指若干个有限幺半群形成的族在态射象、子幺半群、有限直积下封闭)之间存在一一对应。

Büchi 建立了正则文法、有限自动机与逻辑的关系。Higman,Kundu 分别对识别正则语言的子串、词、字典、前缀、后缀的有限自动机的状态数的上下界进行了研究。

参考文献

1. 陈火旺,钱家骅,孙永强. 程序语言编译原理. 2 版. 北京:国防工业出版社,1984

2. Hopcroft J E, Ullman J D. Introduction to automata theory, languages, and computation. Reading, MA:Addison-Wesley, 1979

3. Gonzalez R C,Thomason M G. 句法模式识别. 濮群,徐凤家,徐光祐,译. 北京:清华大学出版社,1984

(张一立)

zhengju lilun

证据理论(evidence theory) 一种重要的**不确定性推理**方法,用质量、信任和似然等函数及证据区间刻画证据的不确定性,用组合规则综合证据。证据理论也称为登普斯特-谢弗理论(Dempster-Shafer theory),简称 DS 理论。

证据理论最初由 Arthur P. Dempster 在 1968 年提出,他试图用概率区间而非单一概率数值去建模不确定性。1976 年,Glenn Shafer 在《证据的数学理论》一书中扩展和改进了 Dempster 的工作,完善了证据理论。证据理论具有好的理论基础,能有效处理"无知(或曰不知道)",避免使用先验概率,能有效综合不同证据源的证据,在专家系统和信息融合等领域得到广泛应用。

在证据理论中，全集 Θ 表示表示环境，Θ 的每个子集表示一个假设，对应一个问题的回答。环境的幂集合 2^{Θ} 包含了所有可能提问的正确答案。证据理论用质量函数(mass function)表示对假设的信任度。该函数定义为从环境幂集合到[0,1]区间的一个映射。空集合的质量值通常被定义为0，环境幂集合所有子集的质量和为1。

证据理论和概率论的基本区别是关于无知的处理。在没有先验知识的情况下，概率论按照中立原理为每个假设分配一个等量的概率值。而证据理论不要求必须对无知假设赋以信任值，而是仅仅将信任值分配给希望对其分配信任的假设。任一未被分配给假设的“信任”被看成“未表达意见”，并将其分配给环境。

当新的证据可用时，希望组合当前所有证据以产生一个更好的信任评价。证据理论采用 Dempster 组合规则组合不同证据源的证据，其一般形式为：

$$m_1 \oplus m_2(Z) = \frac{\sum_{X \cap Y = Z} m_1(X) \times m_2(Y)}{1 - \kappa}$$

式中，m_1 和 m_2 分别表示两个独立差错的证据对应的质量函数，$\kappa = \sum_{X \cap Y = \varnothing} m_1(X) \times m_2(Y)$ 。

κ 值指出了被组合证据相互冲突的程度。当 $\kappa=0$时，两个证据完全相容；当 $\kappa=1$ 时，两个证据完全冲突；当 $0<\kappa<1$ 时，两个证据部分相容。

证据推理采用证据区间[Bel,Pls]表示假设的不确定性，其中区间下界 Bel 被称为支持(support)或信任(belief)，是对假设的最小信任度，区间上界 Pls 被称为似然(plausibility)，表示对假设的不反对程度，是对其的最大信任度。基于质量函数，假设 X 证据区间的下界和上界函数分别定义为：

$$Bel(X) = \sum_{Y \subseteq X} m(Y)$$

$$Pls(X) = 1 - Bel(\bar{X}) = 1 - \sum_{Y \subseteq \bar{X}} m(Y)$$

当被组合的证据存在冲突时，即 $\kappa=1$ 或者接近1时，Dempster 组合规则会产生与直观相反的结果，这是证据理论面临的主要困难。此外，证据间的“独立差错”假设在实际问题中往往难以满足。

围绕基本证据理论，研究者进行了多方面的扩展研究，主要包括：特殊证据函数的计算；证据理论中的近似计算方法；处理证据引起的证据函数的动态变化；证据理论中的非单调关系；利用证据的独立性简化计算；有效组合存在冲突的证据或不满足“独立差错”假设的证据；揭示证据理论与其他不确定性推理模型的关系等。我国学者在证据理论研究方面做出了重要贡献。例如，提出的广义证据理论将有限的环境幂集合推广到一般的布尔代数；提出的简化证据理论在特定领域中将经典证据理论的计算复杂性从指数阶降低到线性阶；提出的凸函数证据理论可有效处理有序命题类问题。

进一步拓展证据理论及其应用范围是其主要的发展趋势。例如，如何有效解决海量、多源、异构信息和知识的融合是证据理论研究面临的一个重要问题。

参考文献

1. Dempster A P. A generalization of Bayesian inference. Journal of the Royal Statistical Society, Series B, 1968, 30(2): 205-247

2. Shafer G. A mathematical theory of evidence. Princeton: Princeton University Press, 1976

3. 刘大有，等. 知识系统中不确定性和模糊性处理的数值方法. 长春：吉林大学出版社，2000

4. Halpern J Y. Reasoning about uncertainty. Cambridge, MA: MIT Press, 2005

(刘大有　杨博)

zheng zhongji

帧中继(frame relay)　在开放系统互连 **OSI 参考模型**七层网络体系结构中第二层数据链路层中，通过逻辑链路的复用和转接以实现以帧为单位的交换称为帧中继交换。这是在数据链路层实现网络资源统计复用的一种快速分组交换技术。在20世纪80年代初帧中继交换技术由X.25分组交换技术演进而来。

在开放系统互连模型中，帧中继与X.25的比较如图1所示：

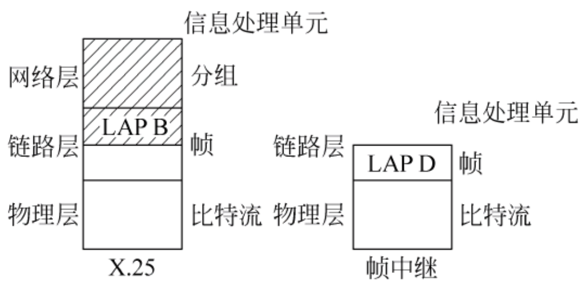


图1　帧中继与X.25比较

帧中继协议只有物理层和数据链路层两层协议。由于物理传输线路采用了光纤，其误码率低、传输速率高。数据链路层协议只相当X.25的数据链路层协议的部分功能。在数据链路层的高级数据链

路控制规程的帧格式中，采用了 ISDN 的 D 通道的规程 LAP D。使帧中继能够在链路层实现逻辑链路的复用和转接，而 X. 25 则是在网络层实现复用和转接，相比帧中继可实现更高的传输效率。

以帧中继交换技术组建的数据交换网称为帧中继数据网。它具有较高的吞吐量，能够提供 1.544 Mb/s 或2 Mb/s，34 Mb/s（或 45 Mb/s）的传输速率。网络延迟很小，帧中继交换机的延迟可小于 2 ms，而 X. 25 分组网延迟为 5 ~ 10 ms。帧中继数据网 20 世纪 80 年代广泛应用于广域网，实现局域网之间高速远程互连，也用于 X. 25 分组网结合，作为 X. 25 分组网的中继网。我国在 1997 年建成并开通帧中继/ATM 数据网（CHINAFR/ATM）。近年来帧中继已经逐步被互联网取代。

参考文献

杜治龙. 分组交换工程. 北京：人民邮电出版社，1993

（马妍　马严）

zhichi guocheng

支持过程（supporting process）　有关各方按他们支持的目标负责的一系列相关活动。支持过程有助于系统或软件产品质量的提高，有助于它们的顺利运行。这类软件过程可被其他类软件过程或本类中的其他软件过程所使用。软件过程的各活动均可使用支持过程。支持过程可由使用它们的机构来实施；或作为一种服务，由一个独立的机构来实施；也可作为项目的一项规定内容，由客户来实施。一个支持过程中的活动，由实施该过程的机构负责。这类软件过程包括：文档过程、配置管理过程、质量保证过程、验证过程、确认过程、联合评审过程、审计过程、问题解决过程等。

文档过程　一个记录由某一过程或活动所产生的信息的过程。这一过程的作用是计划、设计、开发、制作、编辑、发行和维护各类文档。这些文档为系统或软件管理者、工程师以及用户等所必需。文档过程由以下一些活动构成：①过程的准备与实施，②设计与开发，③制作与发行，④维护。

配置管理过程　一个应用配置管理与技术步骤来完成下列工作的过程。这些工作包括：确定、定义一个系统中的软件配置项和基线；控制配置项的修改与交付；记录和报告配置项的完成情况和修改请求；保证配置项的完整性、相容性和正确性；以及控制配置项的存储、处理和提交。配置管理过程由以下一些活动构成：①过程的准备与实施，②配置的确定，③配置的控制，④配置情况报告，⑤配置的评价，⑥发行管理与提交。

质量保证过程　一个为使软件过程和软件产品符合所规定的需求，并按所制订的计划完成提供适当保证的过程。为了避免产生偏见，实施质量保证的人员不能是直接负责软件产品开发的人员，并应在组织上给予独立的权限。质量保证过程由以下一些活动构成：①过程的准备与实施，②软件产品的质量保证，③软件过程的质量保证。

验证过程　验证过程的目的是确定一个系统或软件的需求是否完备和正确，以及每一阶段的软件产品是否达到了前面各阶段对它提出的要求或条件。验证可以和使用它的过程（如供应过程、开发过程、运行过程或维护过程）结合在一起实施，也可由一个独立的机构来实施。验证过程由以下一些活动构成：①过程的准备与实施，②验证。

确认过程　确认过程的目的是确定需求和最终建成的系统或软件是否满足原计划的特定应用。确认可由一个独立于供应人员、开发人员、操作人员或维护人员的机构来执行。确认过程由以下一些活动构成：①过程的准备与实施，②确认。

联合评审过程　联合评审过程的目的是评价项目的某个活动或阶段的执行情况和产品是否合适。它可以由任意二个合作伙伴所使用，由其中的一方评审另一方。联合评审过程由以下一些活动构成：①过程的准备与实施，②项目管理评审，③技术评审。

审计过程　审计过程的目的是确定遵照需求、计划和合同的程度。审计可由任何二个合作伙伴使用，由其中一方审计另一方的软件产品或活动。审计过程由以下一些活动构成：①过程的准备与实施，②审计。

问题解决过程　一个用于分析和排除在开发、运行、维护或其他过程中发现的问题或不一致（不管其性质和来源）的过程。其目的是提供一种适时的、可信赖的并编成文档的手段，以保证分析和排除所有的问题并指明各种倾向。问题解决过程由以下一些活动构成：①过程的准备与实施，②问题解决。

参考文献

1. IEEE Standard for Developing Software Life Cycle Processes — IEEE Std. 1074—1991. 1991

2. ISO /IEC 12207:1995 Information Technolo-

gy— Software — Part 1: Software Life-Cycle Process. 1995 （黄嘉启）

zhichi xiangliangji

支持向量机(support vector machine) 一种监督**模式识别**和**机器学习**方法。采用最大分类间隔准则实现有限训练样本情况下推广能力的优化,通过核函数间接实现非线性分类或函数回归。支持向量机通常简写作 SVM。

基本的支持向量机是针对监督模式识别中的两类分类问题提出的,目的是样本分类与特征向量具有存在但未知的依赖关系的前提下,通过用已知类别标号的样本进行训练,得到能够将样本进行分类的分类器,使其在类别未知的样本上能够尽可能正确地预测分类。原始的 SVM 是 Vladimir Vapnik 提出的最优分类超平面,它定义两类训练样本中离分类超平面最近的样本到分类面的距离为分类间隔,当样本在特征空间中线性可分(存在线性分类函数能够将所有样本正确分类)时,通过最大化分类间隔求得最优分类器。其中,两类中离分类面最近的样本称作支持向量(support vectors)。统计学习理论说明了这种分类器设计方法实际是最小化分类决策函数集的 VC 维,因而在有限训练样本下具有最优的推广能力。

目前通用的支持向量机方法都是基于 Corinna Cortes 和 Vladimir Vapnik 在 1995 年发表的方法,曾经被称作支持向量网络。它通过引入软间隔目标函数实现在样本线性不可分情况下的分类。软间隔目标函数一方面考虑了对于正确分类的样本来说最大化分类间隔,另一方面考虑了最小化训练样本中被错误分类的样本数。

支持向量机对样本的运算只涉及向量间的内积,可以采用核函数内积间接实现非线性变换,实现非线性的支持向量机。由于采用了最大化分类间隔,可以使非线性变换对应的高维空间的 VC 维得到有效控制,保证了支持向量机在采用非线性核时仍然具有好的推广能力。核函数选择需要满足所谓 Mercer 条件,不同类型的核函数会得到不同形式的决策函数。

将支持向量机的输出改为连续变量,用拟合误差替代分类错误率,用离回归函数最近的样本到回归函数的距离作为间隔,就得到用于函数回归的支持向量机,也有人称作支持向量回归(support vector regression 或 SVR)。用于回归的支持向量机同样有线性和非线性回归两种情况。

支持向量机有很多变种,包括 LS-SVM、vSVM、C-SVM 等。在模式识别和机器学习中,采用核函数变换和最大间隔思想还发展了一系列其他方法,通称作核函数机器(kernel machines)。支持向量机在基因表达数据分析、文本识别、网络信息分析、人脸检测与识别等多个领域有广泛的应用。常用的支持向量机软件包有 libSVM、SVMTorch、SVMlight 等。

参考文献

1. Cortes C, Vapnik V. Support-vector networks. Machine Learning, 2005, 20(3): 273-297

2. Vapnik V. 统计学习理论的本质. 张学工,译. 北京: 清华大学出版社,2000 （张学工）

zhishibiaoshi

知识表示(knowledge representation) 在给定表示的形式下,对某个具体事物内涵的描述。知识表示是现实世界到某种抽象形式的镜像。表示(representation)是描述某种事物的形式,是人们描述实际世界的某种抽象语言。知识表示是使用这种语言对这个实际世界的具体描述。

事实上,对实际世界中某个事物的信息处理,往往需要一种简洁且便于信息加工的形式。例如,对物理现象,一般采用数学形式来刻画,这就是一种表示。对统计机器学习来说,在独立同分布(iid)条件满足的条件下,将样本集描述的实际问题,采用一类基函数逼近,然后,设计一种算法,获得一个模型,使得其对样本集的误差小于一个常数。其中基函数就是对实际问题的"表示",模型就是这个实际问题的知识表示。这些事实似乎说明,知识表示是一个顺理成章的问题,没有必要设立专题研究。

人工智能为什么将知识表示作为其基本问题呢?这与人工智能的假设有关:智能的体现,在于有效解决困难问题,其关键是减小搜索空间,为此,不惜使得所使用"模型"对被求解问题不一定完全为"真"。这是智能的特点之一,例如,常识知识与经验知识就具有这种性质。这类知识一般不能使用一类简单函数描述,需要建立变量之间的关系(结构)。这个假设的本质是:为了达到搜索有效,使得模型不绝对为"真"(不确定性),以此换取求解困难问题仅需较小的搜索空间(灵活性)。然而,这是一把双刃剑,提高效率的同时,将会导致例外,即存在很多这个模型不能概括的事例。为了获得有效求解过程,人工智能需要对变量之间的关系作细致描述,

这种描述一般不简洁,需要在结构描述与管理上作出研究,这就是知识表示作为单独问题进行研究的原因之一。另外,人工智能强调知识需要借助结构来表示,因此,在问题求解时,推理成为必然,但是,推理之后,需要对问题求解的过程做出显现解释,以便在模型对某个事例不为真时,能够回答"为什么不为真",这仅当问题被表示为某种显现结构时才有可能。

然而,人们至今不能完全接受人工智能的这个假设,而更加喜欢对某个问题为真的模型(知识表示),这是科学技术自然的追求,这就是为什么强调泛化能力的统计机器学习受到关注的原因。

知识表示研究的另一个问题是试图回答"什么是经验知识,什么是常识知识",这是一个更为本质的问题了。

在人工智能发展的初期,研究者并没有严格区分"表示"与"知识表示",直到 1984 年,Winston 重版他的著作《Artificial Intelligence(人工智能)》时,才将其严格区分为两个概念。

在人工智能中,经常使用的知识表示方法列于表 1。

表 1　各种知识表示方法

表示方法	科学来源
逻辑表示	哲学家与数学家对智能行为中演绎推理作用的研究
产生式系统表示	心理学对智能行为激励-反应现象的刻画
语义网络表示	心理学对记忆的研究
框架表示	对人类感知现象的研究
人工神经网络表示	神经科学研究
概率图表示	图论与统计学
基因表示	生物学对遗传与进化的研究

这些知识表示方法几乎均来源于研究者对智能行为在微观与宏观的不同科学层次之观察与分析所抽象出的形式。对于这些知识表示方法的细节及其优缺点可以参见其他有关条目。

表 1 所列知识表示方法,大致可以划分为两大类:**陈述性表示**和过程性表示。陈述性表示是由一组变量关系和对变量的赋值构成,具有局部含义。它们不随系统其他部分的改变而改变,这类表示包括**逻辑表示**、**产生式表示**、**语义网络表示**、**框架表示**等。**过程性表示**是将实际问题表述为一组可计算的基函数,而基于此类表示的知识表示则是使用某种方法从基函数建立的模型,这类表示包括人工神经网络表示与基因表示等。近年,由于单纯基于统计优化的人工神经网络表示方法遇到困难,人们开始新的尝试,这就是概率图表示。其本意是将结构与统计结合,由此构成一个既可以表示实际问题结构又可以兼顾泛化的模型(知识表示)。

采用逻辑作为刻画智能行为模型的原因在于假设推理是智能的核心,推理是区别智慧与本能的标志。在人工智能中,传统逻辑表示,一般是指一阶谓词原型,其优点是这种知识表示有坚实的数学基础,它保证了推演的保真性及完备性;另外,这种知识表示方法很容易在计算机上实现。但是,它对智能行为的描述,需要满足所建立的模型是对世界某个部分的真实描述的条件,这样才能保证基于逻辑方法所推演的结果与实际现象相一致。这意味着采用这种知识表示方法的任一系统,如果对世界的描述存在误差,将可能导致错误甚至荒唐的结果。此外,一般来说,这种知识表示方法还将遇到计算复杂性方面的困难。

与逻辑表示方法最相近的是产生式系统表示方法(参见**产生式表示**),但是这种知识表示方法无论是从科学观察上还是形式化基础上均与逻辑表示不同。心理学家发现智能行为中的激励-反应现象与产生式原理相一致。这种知识表示原型借用了逻辑蕴涵的操作,但作了适合描述激励-反应现象的转义——符号替代。从形式化角度来讲,在不考虑不确定因子的情况下,这种知识表示方法是保真的,也是完备的。一般地说,在人工智能研究中,为了减小搜索空间或描述可信程度,基于产生式的系统均需要考虑不确定推理,其实质是放弃了逻辑意义下的蕴涵操作,而给予其新的解释,这时它不仅是不完备的而且是不保真的。

语义网络与框架两种知识表示方法产生在不同的年代。20 世纪 70 年代末,它们之间的类同性被说明。这两种方法均适于描述那些具有组织结构的知识,在"一般"与"特殊"之间建立关系,从而,任何对知识特殊性的描述均继承其一般性描述的性质。继承性是这类表示方法的重要特性,并将其作为推理的主要机制。这个特性不仅对人工智能研究起着重要作用,而且为**软件工程**研究开创了一种新的程序设计风格——面向对象的程序设计风格。当然,这里强调的是"风格",而不是语言形式。框架表示方法还有其特殊贡献,其中的"缺省(defaut)"概念,

已成为常识知识表示的重要研究对象。

以上介绍的是人工智能中的几种最基本的陈述性表示方法。它们的共同特点是要求满足“知识显现表示”的条件。换句话说,在系统中,无论是符号还是符号之间的关系均需要显现地说明,即使不确定因子也不例外。这些不确定因子需要具有独立含义:或是一种序或是某种统计分布,这个特性使得基于这些原理建立的模型不具有泛化能力。为此,20 世纪 60 年代被搁置的**感知机**思想,在 80 年代发现了新的学习算法之后,又重新回到人工智能的研究领域。

人工神经网络表示认为,知识取决于一组互相关联的数值,它们当中任一个对所表示的知识均无独立意义。具体地说,知识被分布地表示在这一组数值之中,即以**人工神经网络**作为原型的知识表示方法。在人工智能研究中,这种知识表示方法可说是历经磨难。它的出现早于上述很多知识表示方法将近 20 年,但在 20 世纪 60 年代末,由于寻找其有效学习算法的失败及对其泛化能力的怀疑,导致这种知识表示方法退出人工智能的研究主流,直到 80 年代,人工智能研究者才将这类知识表示方法的原理以连接机制这个术语重新纳入人工智能研究的范围。当然使其重新为智能研究所关注的导火索是反向传播(BP)算法。但是,真正使之成为以后 20 年人工智能主流研究的原因恰恰是:一方面,在理论与算法基础上,Vapnik 基于感知机的支持向量机和由其派生的最大边缘算法,以及 Boosting 等扮演了更为重要的角色;另一方面,基于这类方法成功实现了字符识别,以及分析处理数字网络数据的需求等,这成为这类表示研究的重要推手。

过程性表示方法是人类历史上使用最悠久的传统知识表示方法,数学公式等最经常使用的形式均可归入这类表示。在人工智能历史上关于知识是过程的还是陈述的,有过一段有趣的争论,其本质涉及人工智能最核心的问题,即符号分析与处理在计算机科学中的作用。目前,人工智能研究则将两者考虑为不可或缺的成员,只是不同的研究者或不同的应用领域对它们各有侧重。

由于基于人工神经网络表示的方法与统计学优化方法紧密相关,其关键是将观测样本理解为由所有变量张成的空间上的一个点,由此,学习就是在这个空间上的建模。如果变量个数巨大,这个空间的容量将大到难以想象,无论从统计基础(n 重积分的统计分布计算)还是获得合适样本而言均成为难题,这就是大规模变量集合的统计学在理论上遇到的困难,这使得单纯使用这种表示方法的研究遇到很大的障碍。人们开始找回在人工神经网络发展初期同时萌发的另一类研究,即概率图表示,或者直接称其为**概率图模型**。

概率图表示起源于 20 世纪 80 年代,当时发展了两类模型:其一,**贝叶斯**(Bayes)**网**,这是一类有向图,使用图结构以及变量之间的条件分布,根据特定查询计算某个变量的边缘分布,以作为查询问题的解答;其二,Markov 随机场,这是一类无向图,其原理基于 Gibbs 分布,将无向图中的所有 Clique(或其他定义)构成图,由此根据 Gibbs 分布计算其边缘分布作为特定查询的解答。这两类表示方法在 20 世纪 80 年代在与人工神经网络的竞争中被搁置了,究其原因,主要是学习结构太困难,而使用先验知识(Bayes 网需要先验结构与分布,Markov 随机场需要先验结构)在当时已为研究者所厌恶。这时,无须先验知识的人工神经网络表示更具吸引力,并由此起了主导作用。直到 21 世纪初,当来自人工神经网络的统计机器学习遇到困难之时,这类表示才被重新认识。这类表示的关键是变量之间的相对独立。

对给定变量集合,由这些变量构成一个相互连接的图,如果所有变量两两相连或者完全不相连,可以理解为是一个平凡图,而一个带有非平凡知识的图是,其中部分变量相连,其他不相连。因此,这类表示的关键是找出所有不相连的变量对,即根据变量之间统计条件独立,来决定变量构成的图上,哪些变量之间是不相连的,称为“相对条件独立”集合。这就是概率图表示的要点。目前,这类表示越来越受到研究者的推崇。

上述各种表示方法,仅仅可以理解为描述实际问题的语言,人工智能的研究并不满足于此,它更加关注人工智能自身问题如何表示,这就是经验知识与常识知识。这类知识与理性知识有很大区别,它不是对某个实际问题的精确概括,但是,它在解决问题时,却扮演快速寻找解答,甚至类似顿悟的作用。换句话说,这类知识在搜索解答时,起着启发式知识的作用,它能够大大缩小搜索空间。

根据人工智能研究,特别是在**专家系统**研究中所存在的系统脆弱性等问题,人们认识到其根本原因是常识知识问题未能得到解决。当研究者试图解决这个问题时,对人工智能研究,特别是对知识表示的研究产生了深刻变化。“常识知识存在简洁表示模型吗?”“人工智能需要这类表示常识知识的简洁

模型吗?”“如果这类简洁表示模型存在,是可计算的吗?”对这些问题的回答将涉及“什么是知识表示的本质”这一基本问题(尽管我们已经作了大量讨论,但仅仅涉及如何表示的问题,以及在模型(泛化)意义下的知识表示问题)。根据对这个基本问题的不同理解及所采用的不同方法论,人工智能学界形成了不同的学派。

认识论学派:知识表示是对自然世界的描述,其唯一的作用就是携带知识内容,这意味着知识表示可以独立于推理与搜索来研究,知识表示自身不显示任何智能行为。表示研究与启发式研究无关。

本体论学派:知识表示是对自然世界的一种近似,它规定了看待自然世界的方式,即一个约定的集合。这意味着,对自然世界而言,基于本体论学派,表示只是描述了在这个世界中,观察者当前所关心的那部分,其他部分则被忽略。

知识工程学派:知识表示是计算机对自然世界描述的模型,它应该满足计算机这一实体的具体限制,因此,知识表示可以理解为一类数据结构及在其上的一组操作。

不同的学派将决定智能模拟研究所采用的策略及机器智能行为的评价标准。例如,知识工程学派强调自然世界在计算机内部某类数据结构的映射形式及对存储于其中的内容所采用的处理方法,因此,研究知识的存储结构及对它的有效使用(推理)成为这个学派研究的主要任务。认识论学派认为,知识表示是一种携带知识的理论,至于它是否可计算甚至存在着一个算法均不是重要的,特别是问题求解的有效性完全不在其考虑之列。本体论学派则认为任何知识表示均是不完全的知识理论,而对其使用的有效性(计算困难程度)则是先决条件,因此,本体论学派强调一种聚焦功能,即认为观察者在任一瞬间所关心的事物仅仅是自然世界的一部分,而世界的其他部分则被忽略。这个学派建议将知识表示与计算困难程度联系起来考虑,因此,推理与搜索成为知识表示问题研究的一部分。一般而言,认识论学派强调某种存在性的研究,本体论学派则更多考虑构造性的研究,而知识工程学派则以可实现性作为重点。显然,对任一门学科,存在性、构造性及可实现性均是重要的,简单地否定某个学派是不合适甚至错误的,当然,在某个时期,某个学派主导研究也是自然的。

参考文献

1. Barr A,Feigenbaum E A. The handbook of artificial intelligence. Vol. 1. Pitman, 1981

2. Ginsberg M L. Reading in nonmonotonic reasoning. Morgan Kaufmann Publishers, Inc., 1987

3. Davis R, Shrobe H, Szolovits P. What is a knowledge representation? AI Magazine, 1993, 14(1): 17-33

(王珏　石纯一)

zhishi chanquan he

知识产权核(intellectual property core, IP core)　经过硅片验证的、可重用的、具有特定功能的集成电路模块。简称 IP 核。通常以软核(soft IP core)、固核(firm IP core)和硬核(hard IP core)等形态表现。

软 IP 核以**硬件描述语言**编写的文本形式出现,设计周期短,投入少,灵活性和适应性好;硬 IP 核以掩膜(与工艺相关)形式出现,针对特定工艺进行了性能、功耗和面积上的优化,更易于实现知识产权保护,但可移植性差;固 IP 核以门级电路网表的形式出现,介于软核和硬核之间,在软核设计基础上,完成了门级电路综合和时序仿真等设计工作,保密性和灵活性得到了适当兼顾。

集成电路的复杂度每年以 55% 的速率递增,而设计能力每年仅提高 21%。为缩小二者之间的剪刀差,从 20 世纪 90 年代中后期出现的**片上系统**(SoC)设计方法学引起了工业界和学术界的极大关注,使集成电路(IC)设计开始进一步分工、细化,出现了 IP 核设计和 SoC 系统设计的专业化分工。SoC 设计者通过重用 IP 核,可以在规定的时间周期内研发出复杂的芯片,快速满足市场的需求。IP 核重用已经成为 SoC 设计方法的关键所在。经过 10 多年技术发展,业界已积累了丰富的 IP 核,常用的 IP 核有 CPU、DSP、DRAM、SRAM、EEPROM、Flash、A/D、D/A、MPEG/JPEG/H. 264 编解码、MAC、USB、PCI、SPI、UART、CAN、PWM、RTC、LCD 及各种专业 IP 等,品种越来越丰富,有数千种之多。

IP 核是 SoC/SoPC(可编程的片上系统)设计的基石,其中,SoC 是基于 ASIC 的,SoPC 是基于 FPGA 的。IP 核设计流程一般遵循可重用标准,涉及许多方面,如系统级设计、结构设计、编程实现、验证以及文件编制和可交付清单等,为 SoC/SoPC 设计各阶段提供规则、指南和接口方法,以便可重用 IP 核快捷地、即插即用地集成到 SoC 解决方案中。IP 核一般来源于芯片设计公司的自身积累、Foundry 的积累、专业 IP 公司、EDA 厂商、设计服务公司、IP 资源库

等。IP资源库为IP核建立者和系统设计者提供共享和使用IP核的基础设施。IP核进入该资源库之前,一般先进行IP核认证,进行IP等级分类,让设计者有一个总体概念,如IP核符合标准的准确性程度,再重用需要多大的软插接工作量等。在SoC/SoPC设计过程中对所选的IP核要做适当的优化和系统协调性验证。

IP核和SoC/SoPC相互依存发展,SoC的快速发展离不开IP核的贡献,IP核的发展需要SoC的需求牵引。未来相当长的时期内,IP核和SoC设计方法将推动着集成电路产业的技术的发展。

参考文献

1. 郭炜,等. SoC设计方法与实现. 北京:电子工业出版社,2007

2. 张志敏. 基于“聚芯SoC”的嵌入式系统设计. 北京:邮电大学出版社,2006

(张志敏)

zhishi gongcheng

知识工程(knowledge engineering) 研究知识的获取、表示和利用的理论和方法,以及构建基于知识的系统(knowledge-based systems)所需的技术和工具的一门学科。

发展简史

知识工程的研究使人工智能学科发生了重大改变,从探索普遍的思维规律与通用问题求解方法转向利用知识解决特定领域问题的研究。20世纪60年代初,出现了运用逻辑学和模拟心理活动的一些通用问题求解程序,可证明定理和逻辑推理。但这些通用方法难以解决实际问题,其主要原因在于求解问题需要的巨大搜索空间。实际上,人们解决问题时,常用一些不精确、不确定的启发式规则,以得到有用的结论。1965年,美国斯坦福大学的费根鲍姆(E. A. Feigenbaum)等在总结通用问题求解系统的成功与失败经验的基础上,结合化学领域专门知识,研制了世界上第一个可推断化学分子结构的DENDRAL专家系统(expert system,ES)。**专家系统**是一种最有代表性的基于知识的系统,简称知识系统(knowledge system,KS)。20世纪70年代初美国斯坦福大学研制了MYCIN,是一种帮助医生对血液感染患者进行诊断和选用抗生素类药物进行治疗的专家系统,用LISP语言写成。

1977年,费根鲍姆在第5届国际人工智能联合会议(International Joint Conference on Artificial Intelligence,IJCAI)上做了题为“人工智能的艺术:知识工程课题及实例研究”的特邀报告,首先提出了知识工程概念。费根鲍姆认为,“知识工程是运用人工智能原理和工具解决应用难题的一门艺术,这些问题的解决需要领域专家知识。知识获取、表示,以及恰当运用知识构建和解释推理等是设计知识系统的重要技术。”20世纪80年代初,在日本的第五代计算机研究中,知识工程占有重要的地位。1984年,在微电子和计算机技术公司(Microelectronics and Computer Technology Corporation)启动了由Lenat博士领导的美国Cyc工程(大百科全书计划)。Cyc的主要研究目标是系统获取计算机可理解的人类常识知识,使计算机系统能进行类人推理。Cyc的知识库建设经历了两个阶段:第一阶段是1984年至2004年,主要采用人工方式整理和表示常识知识;第二阶段是从2005年至今,利用Cyc中已有的常识知识,采用自然语言处理的方法,通过阅读文本学习知识。或通过游戏FACTory,吸引志愿者训练Cyc。到2011年,Cyc收集了超过5 000 000条常识知识。

研究内容

知识工程主要研究**知识获取**、**知识表示**、知识推理,以及知识系统开发方法、工具与环境。

知识获取指先从某种信息源(如人类专家、文献、数据等)总结和/或抽取模式,进而将其转换成易于计算机处理的知识,这些知识可用于领域问题求解。知识获取方式通常可分为人工、半自动和自动知识获取三种方式。知识获取是知识系统建造的关键环节,但迄今知识获取任务仍主要通过知识工程师和领域专家的交互来完成,既耗时又低效,成为知识系统构建的瓶颈。事实上,有些专家知识难以用现有方法形式表达,需面向这些专家知识构建一个合适的问题求解模型。这一过程通常要贯穿于知识系统开发、维护的全过程。由于领域知识的复杂性,目前尚无普适、有效的自动知识获取方法。自动知识获取研究主要包括:①对信息源进行泛化、选择、分类和组织;②有效的数据挖掘、机器学习新方法;③知识求精、检查新方法,用于保持知识的一致性、完全性和无冗余性。(参见**知识获取**)

知识表示是为描述知识所作的一组约定,是知识形式化、符号化的过程。不同的应用领域问题,一般需要不同的知识表示方法。理想的知识表示方法应能表示:各种各样的客观事实、各种客观规律和处理规则;事物间结构关系的静态知识,对事物施加各种处理的动态知识;各种精确的、确定的和完全的知识,复杂的、模糊的、不确定的和不完全的知识。

因此,对于一个现实问题能否有合适的知识表示方法往往成为知识处理(领域问题求解)的关键。实际上,一种知识表示方法的优劣对知识处理的有效性及其应用范围的影响很大,对知识获取和学习方法的研究也有直接影响。目前常用的知识表示方法有**产生式表示**、**语义网络表示**、**框架表示**、状态空间、逻辑表示、脚本、面向对象和本体等,以及多种方法结合的知识表示。这些表示方法适于表示各种不同的知识,从而被用于各种应用领域。对简单领域问题,选择一两种现有的知识表示方法就能较好地满足需求。但对复杂领域问题,通常需要选择多种知识表示方法,有时还要提出新的知识表示方法。

知识表示方法与相应的推理方法是密不可分的。前面提到各种知识表示方法,其实都是对应的知识系统的一个组成部分。例如,产生式表示是最常用的一种知识表示方法,而产生式系统则由三部分组成,一组产生式规则、一个数据基和一个推理程序。(参见**知识表示**)

知识推理(推理策略)是知识利用的关键。以采用产生式系统为例,通过反复运用推理控制策略选择合适的规则和证据,以实现问题求解。在演绎推理中常见的推理策略有正向推理、反向推理。正向推理策略可找出其前件与数据基中的事实(或断言、证据等)匹配的那些规则,并用匹配冲突消除策略,从中挑选出一条用于执行,其执行将改变数据基的内容。如此反复进行,直至数据基有与目标一致的事实(即找到答案),或者因无规则可用,或者系统运行超过规定的时间期限,系统将终止运行。反向推理策略是从选定的目标出发,反向使用规则,寻找后件与目标匹配的规则,并用匹配冲突消除策略,从中挑选出一条用于执行;若这条规则的前件与数据基中的事实匹配,问题就得到解决;否则将这条规则的前提作为新的子目标,并对新的子目标寻找可用的规则,直到最后运用之规则前件中的证据可与数据基中的事实匹配,或者直到无规则可用,系统便请求用户回答并输入必需的事实。(建议对其他有代表性的推理方式进行简述)(参见**自动推理**)

20 世纪 80 年代以来,在知识工程的推动下,涌现出了不少专家系统开发工具、平台(或环境),典型的有美国的 EMYCIN、知识工程开发环境 KEE、Cyc,中国的面向对象知识处理系统 OKPS、天马专家系统开发环境和智能化农业信息系统集成开发平台 IPDIAIS 等。知识工程开发环境(平台)和工具显著提高了知识系统开发的效率和质量,促进了知识系统的应用,产生了重大经济、社会和生态效益。

要从根本上解决人类级智能(human-level intelligence)、知识发现和自然语言理解等挑战性难题,除需要自动推理和高效的统计模型之外,还需要以可共享的海量专业知识库和常识知识库——大规模知识系统作为支撑。大规模知识系统是知识经济最重要的基础设施。

海量专业知识库和常识知识库与知识应用中间件(KAPI)相结合,不仅可为知识发现和自然语言理解等系统调用,还能为各类知识服务系统所调用,因此应被视为一种特殊的基础设施。我国学者在 1995 年将其取名为"国家知识基础设施(NKI)",并且于 2000 年正式启动相关的研究工作。

大规模知识系统的代表性研究工作还有基于本体(ontology)的知识管理系统 KMSphere。本体可以有效地进行知识表达、知识查询、语义消解。本体知识管理可实现语义级的知识服务,提高知识利用的深度。本体知识管理还可以支持对隐性知识进行推理,方便异构知识之间实现互操作。

知识工程应用于**决策支持系统**(DSS),产生了智能决策支持系统,由数据库、模型库、知识库、图形库和文本库组成,为解决半结构、非结构化的决策问题提供有效的手段。

今后知识工程研究的重点是知识模型,知识操作语言,专业知识库与常识知识库结合,知识发现、数据挖掘与推理的深度结合,知件等问题。

参考文献

1. 陆汝钤. 人工智能(上). 北京:科学出版社,1988

2. 史忠植,王文杰. 人工智能. 北京:国防工业出版社,2007

(史忠植)

zhishi huoqu

知识获取(knowledge acquisition) 将用于领域问题求解的专门知识从人类专家或其他知识源(如文献、数据库和 Internet 等)中总结和抽取出来,然后转换成一种形式化的知识(如规则、框架等)的过程。

知识获取是建立知识系统的关键,但也是一个耗时、低效的过程,故一直是知识系统建造的"瓶颈"。知识获取主要研究:对人类专家或其他知识的理解、选择、抽取、汇集、分类和组织的方法;从已有的知识和数据中产生新知识,包括从外界学习新知识的机理和方法;检查或保持已获取的知识集合

的一致性、完整性和无冗余性的方法。

知识获取所涉及的阶段可分为：①问题识别阶段：确定参与人员、资源、系统设计目标和问题主要特征等；②概念化阶段：确定领域基本概念及其关系，对系统目标进一步分解等；③形式化阶段：选择合适的**知识表示**方式，将基本概念、关系、子任务等表示出来，确定相应结构和问题求解模式；④实现阶段：将形式化阶段得出的知识映射成可执行的程序，实现一个知识系统原型；⑤测试阶段：选用更多、更全面的例子从表示、推理、结构等方面评价原型系统，对不合适部分进行修改。以上五个阶段是一个不断反复的过程。

通常，知识获取的途径主要有三种，即人工知识获取、半自动知识获取和自动知识获取。具体地说，知识获取方法主要有：①会谈式知识获取，知识工程师与领域专家直接交互以获取知识。②案例分析式知识获取，领域专家以具体案例为线索，进行分析、归纳后给出问题的答案。③教学式知识获取，知识工程师通过知识编译器（专用于知识处理的编译器）给计算机传授知识。④归纳式知识获取，从一些特定案例或不完全的局部事实、关系、概念等出发，归结出带有一般性的推理模式或因果关系。⑤假设式知识获取，根据对外界现象的观察，进行归纳、联想、类比、分析和综合等，形成假设，进一步对假设进行验证。如此反复多次，最终形成满意的结果。⑥基于**机器学习**方法的知识获取，是指系统直接从环境中获取所需的知识。随着机器学习研究的日益深入和大量学习算法的出现，机器学习已成为自动获取知识的强有力工具。其中，代表性的技术有：关联规则挖掘、统计方法、**贝叶斯网**学习、**人工神经网络**、模糊神经网络、**决策树**、粗糙集、**遗传算法**、案例推理、概念格、本体论、粒度计算等。实践表明，将上述某些知识获取方法按照一定的原则结合使用，通常可达到更好的知识获取效果。

随着 Internet 的普及，以及**传感网**、**物联网**、信息检索、信息融合、知识发现等技术的不断发展，知识获取手段也在不断地演化。未来基于 Internet 平台上的海量数据，借助于信息检索、知识发现等技术进行知识获取是一个重要的发展趋势。此外，在传感器网络、物联网等平台上通过信息融合技术来获取知识也是一个非常有前途的研究方向。

参考文献

1. Boose J H, Gaines B R. The foundation of knowledge acquisition. Academic Press, 1990

2. Rhem A J. A framework for knowledge acquisition (White Paper). 2001. http://www.ajrhem.com/framework.pdf

3. Kendal S L, Creen M. An introduction to knowledge engineering. Springer, 2007

（徐从富　何钦铭）

zhishi ku

知识库（knowledge base）　面向应用领域问题求解的需要，将知识用某种（或某些）**知识表示**方法表达、组织、存储在计算机中，便于使用和维护，既相互关联又相对独立的知识片集合。譬如，产生式系统（参见**产生式表示**）中的一条产生式规则就是一个知识片。

20 世纪 60 年代中期，E. A. 费根鲍姆（Edward A. Feigenbaum）等人在建造第一个**专家系统**的同时，也建造了第一个知识库。许多应用程序都利用知识，有的还达到了相当高的水平。但这些应用程序可能并不是基于知识的系统，它们并不包含知识库。一般的应用程序与基于知识系统的区别在于：一般的应用程序把问题求解的知识隐含地编码在程序中，而基于知识的系统则显式表达领域知识，并将它们组成一个相对独立的实体（见图 1）。

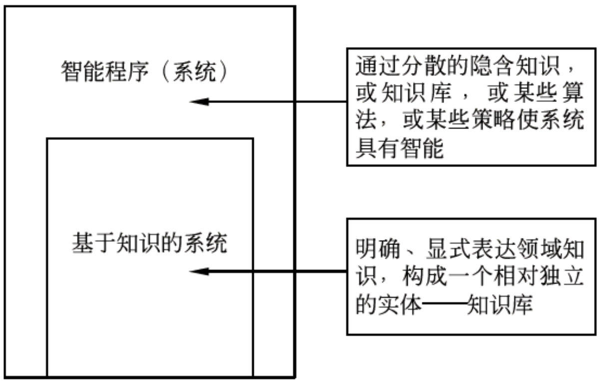


图 1　基于知识的系统与一般智能程序的区别之一

知识库便于领域知识的检索、共享和管理，是基于知识系统（如专家系统）的重要组成部分。知识库是决定专家系统智能水平和成功与否的关键因素之一。知识库的构造必须使得其中的知识在使用的过程中能够被有效地存取和搜索；库中的知识能方便地被修改和编辑；同时，对库中知识的一致性和完备性能进行检验。

知识库具有如下特点：

（1）知识库中的知识根据它们的应用领域特

征、背景特征(获取时的背景信息)、使用特征、属性特征等形成便于利用、有结构的组织形式。知识一般是模块化的。

(2) 知识库中的知识通常分为两类,一类是常识知识,一类是领域知识;因此知识库也分为常识知识库和领域知识库。

(3) 知识库中的知识可由多种知识表示方法进行表达。如常识知识和领域知识可由规则、语义网、框架和一阶逻辑等方法表示。

(4) 知识库中还可存在一个通常被称作"典型方法库"的特殊部分。如果解决某些问题所需的事实或规则等是确定的,可将与这些问题对应的信息直接存储在"典型方法库"中。在问题求解时,如果遇到"典型方法库"中存储的某问题,有关该问题的事实和规则将被调出,直接生成实际问题的解,无须再从庞大的规则库中搜索相关规则。

当所使用的知识规模庞大或分散在不同的地理位置时,可使用分布式知识库对知识进行有效组织和存储。一方面,可大大降低建造大规模、集中式知识库的成本,另一方面还可提高知识系统的鲁棒性和推理效率。

要从根本上解决**知识工程**领域的许多挑战性难题,构建可共享的海量专业知识库和常识知识库是必要的基础。2000 年开始,我国陆续启动了相关的研究工作,将海量专业知识库和常识知识库与本体、知识应用中间件相结合,应用于知识发现、**自然语言理解**以及各类知识服务系统中。海量、异质、动态信息源中的多领域本体建模,知识获取、表示、融合和组织,本体知识管理等问题都是构建大规模知识库系统的关键问题。

参考文献

1. Russell S, Norvig P. Artificial intelligence: A modern approach. 3rd ed. Upper Saddle River NJ: Prentice Hall, 2010

2. Giarratano J C, Riley G D. 专家系统原理与编程. 4 版. 印鉴,陈忆群,刘星成,译. 北京:机械工业出版社, 2006

3. 管纪文, 刘大有,等. 知识工程原理. 长春:吉林大学出版社, 1988

(刘大有 郑方青 黄晶)

zhijie cunchuqi cunqu

直接存储器存取(direct memory access, DMA) 不经过**中央处理器**、直接用硬件实现的外围设备与**主存储器**之间的高速数据传送。

当某些外围设备,诸如磁盘、CRT 显示器、高速模数转换器等要求高速而大量地传送数据时,采用程序控制方式来传送数据往往无法满足速度的要求,就以程序控制方式(参见**输入输出控制方式**)中传送速度最快的中断方式而言,每传送一个字节(或一个字)就得把主程序停下来,转而去执行中断服务程序,而在执行中断服务程序前要做好现场保护,执行完中断服务程序后还得恢复现场。由于在程序控制方式中数据传送过程始终受中央处理器(CPU)的干预,CPU 都需要取出和执行一系列指令,每一字节(或字)数据都必须经过 CPU 的累加器才能输入/输出,这就从本质上限制了数据传送的速度。为此提出了在外设和内存之间直接地传送数据的方式。即 DMA 传送方式。

实现 DMA 传送可以利用中央处理器在执行与访问主存储器无关的操作时进行,在传送过程中不需要中央处理器介入。此时外存储器窃取了中央处理器的周期,所以这种 DMA 传送称**周期窃取**。在周期窃取期间,程序计数器保持不变,输出输入操作与中央处理器并行工作,主存储器通过输入输出接口直接与外围设备连接。它不用程序而用专门的 DMA 控制器直接实现主存储器与外围设备之间的数据发送和接收,其传送速度仅受存储器的数据通道宽度和外围设备的数据传送特性的限制,它是一种高速数据传输操作,常用于高速外围设备与主存储器、主存储器与主存储器、外围设备与外围设备之间的大量数据块传送操作。

在 DMA 传送数据时要占用**系统总线**。根据占用总线的方法不同,DMA 可分为中央处理器停止法、总线周期分时法和总线周期挪用法等。无论采用哪种方法,在 DMA 传送数据期间,中央处理器不能使用总线。因此,DMA 传送虽然提高了数据传送率,但仍占用中央处理器时间。当然与程序控制方式相比,占用的中央处理器时间缩短了很多。因此,不同的系统都提供了与该系统配套的 DMA 控制器来实现 DMA 传送的管理。

参考文献

1. 金兰,金波. 计算机组织:原理、分析与设计. 北京:清华大学出版社,2006

2. 唐朔飞. 计算机组成原理. 2 版. 北京:高等教育出版社,2008

(徐良贤)

zhijie tihuizhi jishu

直接体绘制技术(direct volume rendering technique) 基于离散的三维数据场数据直接产

生二维图像的一种绘制技术。与等值面方法不同，在这一过程中并不需要产生中间几何图元。体绘制技术的优点是能从所产生的图像中观察到三维数据场的整体和全貌（见图1），而不只是显示出人们感兴趣的等值面（参见**等值面抽取技术**）；同时，体绘制也易于并行处理。

体绘制包括两个步骤：三维数据场的重采样；采样点光亮度贡献的计算。根据不同的重采样方法，体绘制技术可分为图像空间和物体空间两种方法。

图像空间方法逐行、逐点地计算屏幕上每一个像素的光亮度，即从每一个像素发出一条射线，沿射线对数据场进行采样，采样点处的数据值由其邻近体素数据值插值得到（如三线性插值）。继而对数据场的数据进行分类并赋予不同的颜色值及不透明度值。最后，从前往后将每一个采样点的颜色值及不透明度值依次按下列公式迭代合成：

$$\alpha_{out} = \alpha_{in} + \alpha_{now}(1 - \alpha_{in})$$

$$C_{out}\alpha_{out} = C_{in}\alpha_{in} + C_{now}\alpha_{now}(1 - \alpha_{in})$$

式中，C_{in}，α_{in}分别代表射线进入当前重采样点之前的颜色和不透明度；C_{now}，α_{now}分别表示当前重采样点的颜色和不透明度；C_{out}，α_{out}分别表示叠加上重采样点的颜色和不透明度后的相应值。

物体空间方法首先对数据场进行分类，给定各采样点处的不透明度值及颜色值，并将各采样点按其距离视点的远近逐个由物体空间投影到屏幕空间的各像素上，在屏幕空间累计每个像素的光亮度。

直接体绘制的代表性算法分别是图像空间的光线投射法（ray casting）和物体空间的抛雪球法（splatting）。光线投射法绘制质量较高。抛雪球法也称为足迹表法（footprint method）。该方法以物体空间为序，计算每一体素投影的高斯强度分布的影响范围（即足迹，footprint），并加以合成，形成最后的图像。该方法可以只选取与图像相关的体素进行投射和显示，适合稀疏体数据的绘制和物体空间的并行处理。

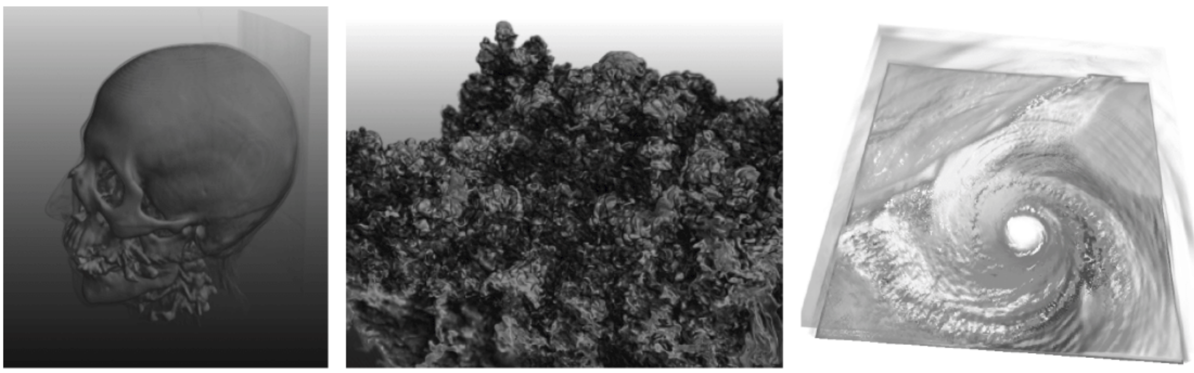
提高体绘制效率的原则是尽量减少不必要的体素渲染数量。近来发展的技术包括空区间跨越、光线早期终止、八叉树和BSP空间分割、自适应多分辨率采样、预积分体绘制等。

随着图形硬件的发展，可将三维体数据场视为三维纹理，装入图形硬件内存，利用硬件实现线性插值及图像合成的并行运算，从而提高体绘制效率。基于三维纹理的直接体绘制方法主要利用硬件的三线性过滤插值能力，通过渲染多个与视线垂直的面片来重建整个三维结构，每个面片利用三维纹理来决定颜色与透明度。这种方法得到的效果与光线投射的效果相同。也可直接利用三维纹理在图形硬件上实现光线投射算法。

在直接体绘制中，需要采用传输函数（transfer function）将三维数据场的采样点映射成光学参数。传输函数的选择直接决定了绘制的效果，成为体绘制研究的关键。然而传输函数的设计比较复杂，现有的算法存在盲目性、用户界面不直观、参数调节复杂等问题。近来对传输函数的研究进展包括高维传输函数、基于特征的传输函数方法等，可更有效地改善绘制效果。**非真实感图形绘制**方法也被用于体绘制。这类方法通过边界增强（boundary enhancement）、侧影轮廓（silhouettes）、网点模式（stipple patterns）等方法，强调关注的三维体数据特征，利用艺术化的方法，获得特殊的可视化效果。

参考文献

1. 石教英，蔡文立. 科学计算可视化算法与系统. 北京：科学出版社，1996



CT数据　　流场模拟数据(涡量)　　飓风模拟数据(多变量标量场)

图1　直接体绘制效果

2. 唐泽圣,等. 三维数据场可视化. 北京:清华大学出版社,1999

3. Wallis J W, Miller T R, Lerner C A, Kleerup E C. Three-dimensional display in nuclear medicine. IEEE Trans Med Imaging, 1989, 8(4):297-303

(袁晓如)

zhijue zhuyi luoji

直觉主义逻辑(intuitionistic logic) 根据以荷兰数学家 J. Browwer 为代表的直觉主义观点建立起来的逻辑。直觉主义者认为:逻辑仅是数学的一部分,决非数学的基础;数学应该产生于直觉,数学的证明只能用有穷方法或构造方法,所谓命题 φ 真是指有 φ 的一个使用有穷方法或构造方法的证明。因此,直觉主义者不接纳实无穷,不承认排中律。他们认为排中律仅在有穷情况适用,并不具有普遍性,因而不能随意滥用。关于逻辑联结词,直觉主义者作了如下解释:

(1) 命题 $\varphi \wedge \psi$ 真,是指有 φ 的一个证明及 ψ 的一个证明;

(2) 命题 $\varphi \vee \psi$ 真,是指有 φ 的一个证明或 ψ 的一个证明;

(3) 命题 $\varphi \supset \psi$ 真,是指有一个证明,它能把 φ 的每个证明转变为 ψ 的一个证明;

(4) $\perp$ 无证明;

(5) $\neg \varphi$ 定义为 $\varphi \supset \perp$。

第一个直觉主义逻辑系统是由 A. Heyting 于 1930 年建立的,称为海丁系统。海丁系统是一个希尔伯特型系统,其命题演算系统的公理如下:

$\varphi \supset (\varphi \wedge \varphi)$

$(\varphi \wedge \psi) \supset (\psi \wedge \varphi)$

$(\varphi \supset \psi) \supset ((\varphi \wedge \theta) \supset (\psi \wedge \theta))$

$((\varphi \supset \psi) \wedge (\psi \supset \theta)) \supset (\varphi \supset \theta)$

$\psi \supset (\varphi \supset \psi)$

$(\psi \wedge (\psi \supset \varphi)) \supset \varphi$

$\varphi \supset (\varphi \vee \psi)$

$(\varphi \vee \psi) \supset (\psi \vee \varphi)$

$((\varphi \supset \theta) \wedge (\psi \supset \theta)) \supset ((\varphi \vee \psi) \supset \theta)$

$\neg \varphi \supset (\varphi \supset \psi)$

$((\varphi \supset \psi) \wedge (\varphi \supset \neg \psi)) \supset \neg \varphi$

古典命题演算中的许多形式定理都仍然是海丁系统中的形式定理,但也有一些古典命题演算中的形式定理,如排中律 $\varphi \vee \neg \varphi$ 不是海丁系统中的形式定理。

在海丁系统出现之后,许多人对直觉主义逻辑进行了更深入的研究,G. Gentzen 给出了直觉主义逻辑的自然演绎系统,S. A. Kripke 在 1965 年提出了一种直觉主义逻辑的语义模型,后人称为克里普克结构,直觉主义逻辑系统关于克里普克结构是完全的。

参考文献

Bell J L, et al. A course in Malhematical logic. North-Holland, 1977

(宋方敏)

zhilian cunchu

直连存储(direct attached storage, DAS) 一种直接通过电缆与计算机相连的存储设备,也称直接附属存储。系统存取访问的输入输出(I/O)请求直接在计算机和存储设备间进行,存储设备由主机独享。

早期存储数据量较少时,在计算机中可能内/外置带有 SCSI 接口的磁盘或磁带驱动器(或**磁带机**),采用直连存储方式实现数据存储。随着数据量的增加以及对数据可靠性的要求越来越高,磁盘阵列(参见**磁盘阵列**)已经成为计算机直连存储的主要设备,并且直连存储还可以构成基于磁盘阵列的双机高可用系统,满足数据存储对高可用的要求。然而,随着数字化进程的不断深入以及网络存储技术的发展,数据存储逐渐独立于服务器,但计算机中系统软件以及一些基本信息仍然采用直连存储方式。目前常用的直连总线有 SCSI、PATA、SATA(参见**外存储设备接口**)等。

直连存储具有性能高、数据安全性好、价格便宜等优点,但由于存储设备直接与计算机相连,这种存储连接方式扩展、维护困难,甚至需要停机进行存储扩展与维护,而且不利于数据资源的共享,造成总体成本较高。

直连存储依赖计算机主机操作系统进行维护管理,占用计算机主机资源,对服务器硬件的依赖性较大。

参考文献

鲁士文. 存储网络技术及应用. 北京:清华大学出版社,2010

(陈俭喜)

zhiliu dianyuan

直流电源(direct current power supply) 为计算机各部分提供直流电能的设备。

计算机的直流电源有磁放大器电源、整流电源、

线性电源、开关电源等，最常见的是线性电源和开关电源。开关电源又可分为串联式开关电源和无工频变压器开关电源两种。

磁放大器电源 以磁性元件作为控制和调整环节的供电设备。磁放大器常用的有扼流圈形式和快速形式等。一般磁放大器电源以快速磁放大器作为电压补偿元件，以晶体管直流放大器作为误差检测和信号放大元件来构成负反馈有差调整系统。磁放大器稳压电源的优点是可靠性高，体积小，不会发生振荡。

整流电源 利用半导体器件或其他方法直接将交流电转换成直流电的装置。自20世纪20年代起至今已经历了旋转式变流机组(电动机-直流发电机)、静止式离子整流器和半导体整流器的几个发展阶段。目前常用半导体整流器构成整流电源，它包括整流二极管和晶闸管(SCR)构成的整流和可控整流电源。

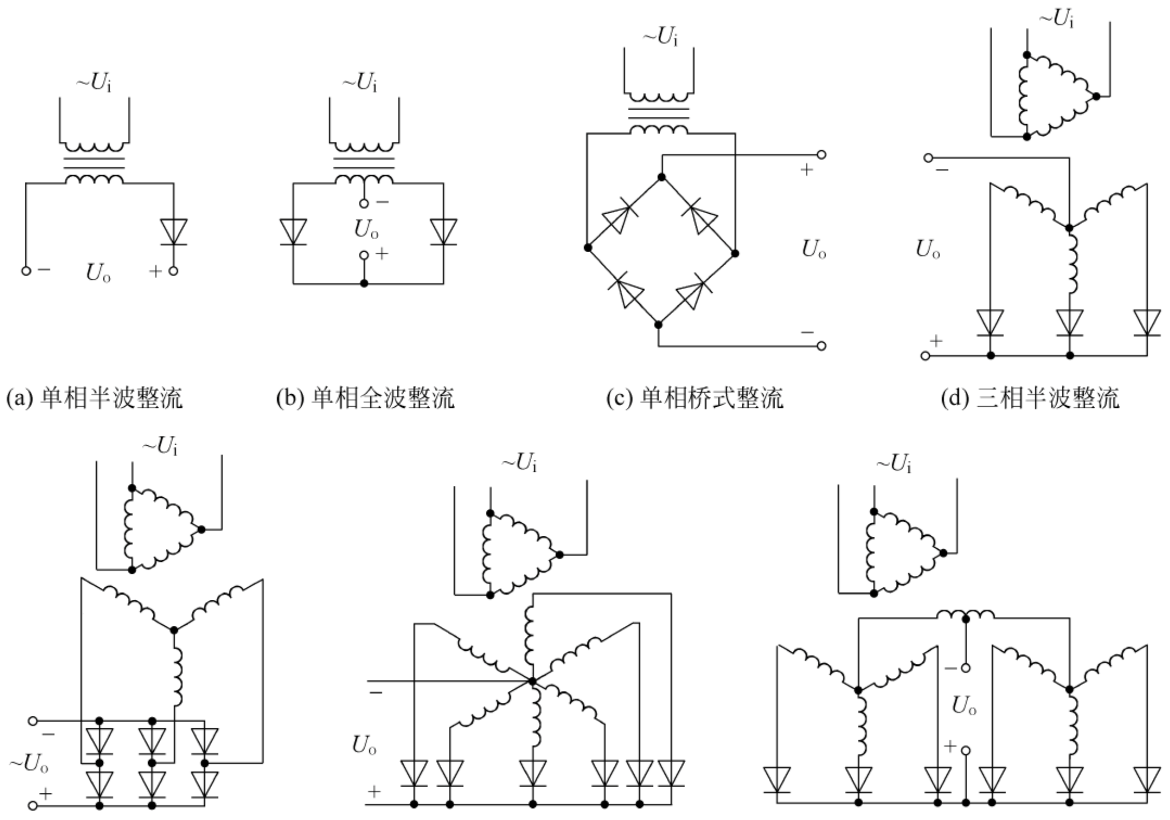
图1是常用的整流电源电路，它是开环的，当输入交流电压 U_i 变化时，输出直流电压 U_o 随着变化。用晶闸管构成的可控整流电源，具有控制简单、可靠性高和输出功率容易增大的优点。整流电源在大、中、小型计算机中都有应用。

线性电源 通过调节与负荷串联的调整管压降或与负荷并联的调整管电流来达到稳定输出电压或输出电流的目的一种供电装置。线性电源又称直流线性电源。它是一个闭环自动调整系统，包括串联方式和并联方式，其控制电路有电压控制和电流控制两种。

图2表示串联式线性稳压电源，它由调整管(通常是功率三极管或达林顿功率管)、采样电路、放大器、基准电压和辅助电源等组成。在电压控制方式的线性电源内调整管工作在功率三极管输出特性曲线的线性区，当输出滤波电容值很小时，电路的动态响应很快。输出电压的纹波也非常小，通常小于1 mV。稳定度可达1% ~0.1%，噪声也很小。

线性电源应用广泛，100 W 以下的低压电源已经集成化，使用非常方便。利用集成稳压芯片和运算放大器等器件组成的稳压电源线路十分简单，它的输出电压纹波小，稳压精度高，且动态响应快。

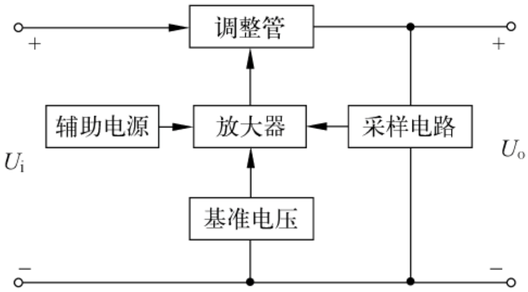
开关电源 通过调节功率器件的开通和截止时间来达到稳定输出电压和电流的供电装置。它是一个非线性闭环调节系统。自20世纪60年代起开关电源发展迅速，是计算机电源最常见的形式。后来



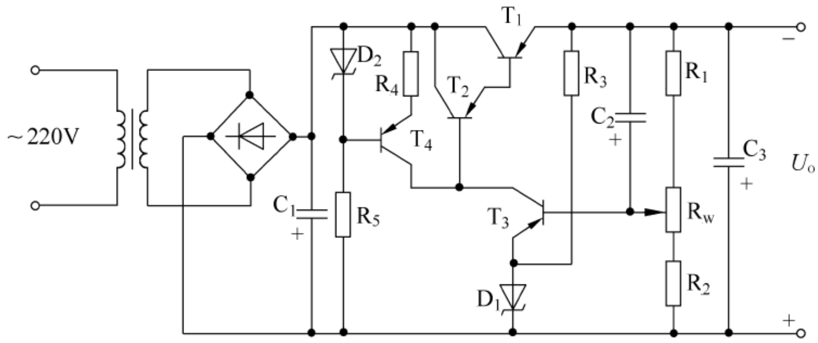
(a) 单相半波整流 (b) 单相全波整流 (c) 单相桥式整流 (d) 三相半波整流

(e) 三相桥式整流 (f) 三相中心抽头整流 (g) 三相双星形整流

图1 典型的单相和三相整流电源电路



(a) 框图



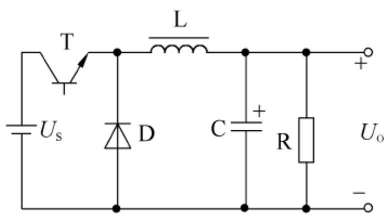
(b) 电路图

图 2　串联式线性稳压电源

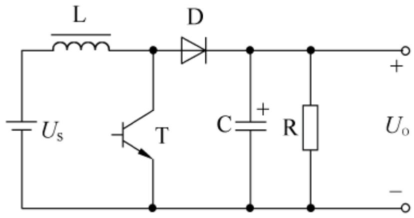
开关电源的功率已达千瓦级。集成化模块电源的输出功率也不断提高,如:输入 48 V、输出 5 V 20 A 模块等产品已上市。

开关电源的基本形式有 4 种:串联、并联、反极性以及古卡,分别见图 3(a),(b),(c),(d)。这些基本形式的前级通常是工频变压器和整流滤波电路。开关电源的控制方式主要有 4 种:①自由振荡型,即开关频率、开通和关断时间均在一定范围内变化;②脉宽调制型,即频率固定,开通或关断时间可变;③谐振型,即开通或关断时间固定,频率可变;④移相型,电路内至少有两个开关器件,同一频率的两列方波之间的相位差可变。

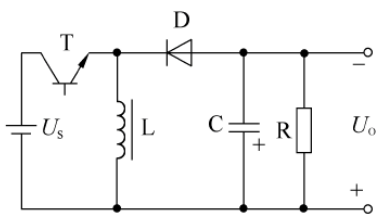
无工频变压器开关电源是用高频变压器代替工频变压器,电源的尺寸和重量显著减小。无工频变压器开关电源的常用形式有正激、反激、推挽、半桥、全桥、古卡和串、并联谐振等。图 4 是典型的无工频



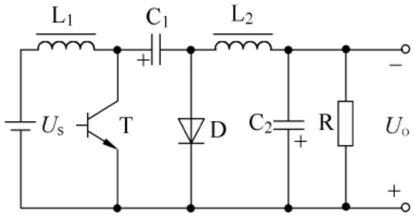
(a) 串联形式



(b) 并联形式



(c) 反极性形式



(d) 古卡形式

图 3　开关电源基本形式

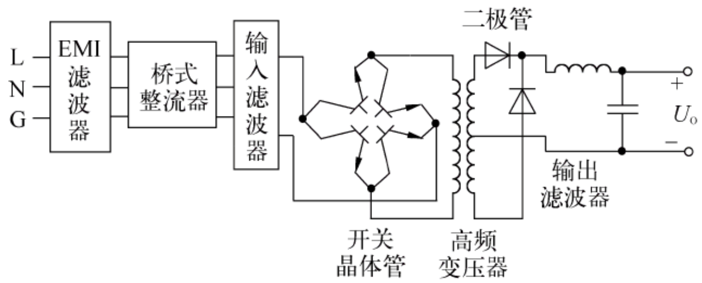


图 4　典型的无工频变压器开关电源方框图

变压器开关电源方框图,它包括抗电磁干扰(EMI)滤波器、桥式整流器、输入滤波器、开关晶体管、高频变压器、次级二极管整流和输出滤波器,控制部分采用线性集成电路。

微型计算机中常用的是无工频变压器开关电源。新的技术,如开关损耗为零、抑制电磁干扰、校正功率因数等,也已得到广泛应用。

三种常见的稳压电源的性能比较见表 1。

表 1　三种稳压电源性能比较

比较项目 \ 电压类型	线性稳压电源	串联式开关稳压电源	无工频变压器开关稳压电源
电压稳定度(电网变化 ±10%)	0.01% ~0.05%	0.1% ~0.2%	0.1% ~0.2%
负载稳定度(负载变化 0 ~100%)	0.02% ~0.05%	0.1% ~0.2%	0.1% ~0.2%
输出纹波(峰-峰值)	<5 mV	10 ~100 mV	10 ~100 mV
上升建立时间	100 μs	100 ms	100 ms
电压维持时间	5 ms	25 ms	25 ms
过渡过程	0.02 ~0.1 ms	0.5 ms 至数 ms	0.5 ms 至数 ms
效率	30% ~40%	40% ~70%	60% ~85%
体积	1	2/3 ~1/3	1/3 ~1/5
重量	1	2/3	1/3
可靠性(MTBF)	100 000 h	50 000 ~100 000 h	30 000 ~100 000 h
控制电路元件个数	10 ~30 个	10 ~50 个	30 ~120 个

计算机的发展对电源装置的性能指标,特别是电源的尺寸、重量、可靠性以及可维性等要求逐步提高,促使电源向高功率密度和高可靠的方向发展。开关电源高频化是电源小型化的重要途径,开关电源的工作频率为 20 ~ 50 kHz,将提高到 100 ~ 500 kHz 以上。频率的提高不仅使电源重量减轻,体积减小,效率提高,动态特性和小信号扰动下的稳定性改善,并且使传导或辐射的电磁干扰容易被滤波、屏蔽和抑制。　(方资端　林兼)

zhi

值(value)　可直接进行计算、储存、作为复合数据的成分,传送作为函数或过程的变元,以及作为函数结果所回送的任何实体。或称为计算中存在的任何实体。

值是计算机科学中最为基本的概念之一。语言不同,值的种类也各异,例如,在 **PASCAL 语言**中可区分如下几类值:初等值(逻辑值、字符、枚举值、整数、实数),复合值(记录、数组、集合、文件),指引元,变量引用,过程及函数抽象;而在 ML 语言中却有如下几类值:初等值(逻辑值、整数、实数、串),复合值(元组、记录、构造、表、数组),函数抽象,变量引用。

参考文献

Watt D A. Programming language concepts and paradigms. Prentice Hall,1990　(徐家福)

zhidu cunchuqi

只读存储器(read only memory,ROM)　一

次写入信息后能长期存储，此后只可读取的一种存储器。在一般情况下，ROM 所存的内容在较长时间内是相对固定的，即使在断电情况下，所存信息仍不会消失，因此 ROM 是非易失性存储器。

早期的只读存储器采用分立元器件，如二极管矩阵、磁心矩阵、变压器和电容器等，后来用半导体集成电路技术制造只读存储器。半导体只读存储器的基本存储模块为只读存储器电路芯片。电路主体是一个由大量存储单元组成的存储阵列，每个存储单元主要由位于阵列字线和位线交叉处的半导体管构成，半导体管通或断分别代表存 1 或存 0。半导体只读存储器写入数据的过程称为对 ROM 编程。根据编程方式的不同，半导体 ROM 可分为三类：①掩模型只读存储器（掩膜 ROM）。这类 ROM 所存的数据在芯片制造过程中完成编程。使用时只能读出，不能写入。②可编程只读存储器（PROM）。这类存储器所存的数据可由用户根据自己的需求进行一次性编程。此后，存储器只能读出，不能写入。③可改写只读存储器。这类存储器有可擦编程只读存储器（EPROM）和电可修改只读存储器（EAROM）两种。这类存储器可用紫外光照射或电的方法擦除已写入的数据，然后再用电的方法重新写入新的数据。用户可以根据自己的需要多次改写 ROM 内容。由于 EPROM 和 EAROM 改写时间比读出时间长得多，有些电路的改写又需在专门的电路上进行，故它们在工作时只读不写，仍属于只读存储器。

只读存储器广泛用作各种类型的固件，如用于存放接通电源后启动计算机的引导程序、高级语言的加工程序和各种微程序。只读存储器经常被做成代码变换器、字符产生器、数学函数表、汉字字型库、游戏卡、音乐卡等，也可实现各种组合电路功能，配套构成时序电路，还可以 ROM 网络的形式实现特殊的逻辑函数需求。由于只读存储器能有效地提高计算机软硬件的功能并且具有可靠性、灵活性及经济性的特点，所以它已成为现代计算机系统的重要部件。

除了半导体只读存储器外，还有一类只读存储器，它们按直接存取方式工作，这就是光盘存储器中的只读光盘存储器（CD-ROM）和一写多读光盘存储器。只读光盘存储器上的信息是母盘经模压复制在光盘上的，只具有读取信息的功能，用户不能改写。只读光盘是由视听光盘演变而来，用在计算机上的 CD-ROM 与 VCD、SVCD、DVD 等家电产品的光盘结构相同，但记录格式不同。CD-ROM 可用以存储各类软件产品，如系统软件、应用软件、专家系统、多媒体信息以及各种电子出版物等。由于光盘容易复制、价格便宜、容易使用，已非常流行。一写多读光盘存储器的写入过程是激光束使存储介质相应部位“烧蚀”，已经永久性地改变了其光学性质，写后不能修改，但可多次读取。它常用以存储文件、档案、资料等。

参考文献

1. 张江陵，金海. 信息存储技术原理. 武汉：华中理工大学出版社，2000

2. Hodges D A. Semiconductor memories. IEEE Press. 1972

（时万春）

zhidu cunchuqi xinpian

只读存储器芯片（read only memory chip）

在正常运行情况下，只有读出操作，没有写入操作的**半导体存储器芯片**。简称 ROM 芯片。写入数据后，即使切断电源，ROM 芯片存储内容仍不会消失，所以是非易失性存储器。随着微电子和存储器技术的发展，习惯上把具有系统运行中写入数据能力，但应用中以读出操作为主的非易失性存储器也视为 ROM 芯片。

ROM 芯片分类　ROM 芯片产品是 1970 年问世的，早期产品是**掩膜型只读存储器**（mask ROM）**芯片**，其存储内容是在集成电路生产过程中由集成电路生产者按用户要求写入的。随后出现了可由用户自行编程，但只能一次性写入数据的 ROM 芯片，即基于双极工艺的高速熔丝型**可编程只读存储器芯片**（PROM 芯片）。1971 年又生产出可由用户编程写入数据并且可擦除的**可擦编程只读存储器芯片**（EPROM 芯片），它是一种采用浮栅雪崩注入 MOS 工艺制造的紫外线擦除的 EPROM 芯片，记为 UV EPROM 芯片。随着系统运行中可编程需求的增长，1977 年研制出可现场修改存储内容的**电可修改只读存储器芯片**（EAROM 芯片），称之为**电可擦编程只读存储器芯片**（EEPROM 芯片）。20 世纪 80 年代以来陆续推出一些新的 ROM 芯片形式，如一次可编程 EPROM 芯片（OTP EPROM 芯片）、**快可擦编程只读存储器芯片**（Flash EPROM 芯片）和非易失性随机存取存储器芯片（NV RAM 芯片）等。图 1 为只读存储器芯片分类。

掩膜型只读存储器芯片　早期的掩膜 ROM 芯片以 NMOS 管作为基本存储单元，后来过渡到以互补金属-氧化物-半导体（CMOS）为主。图 2（a）为一

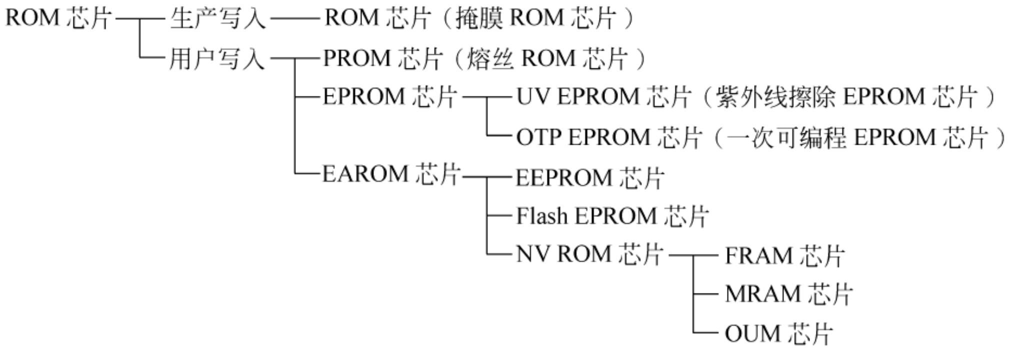
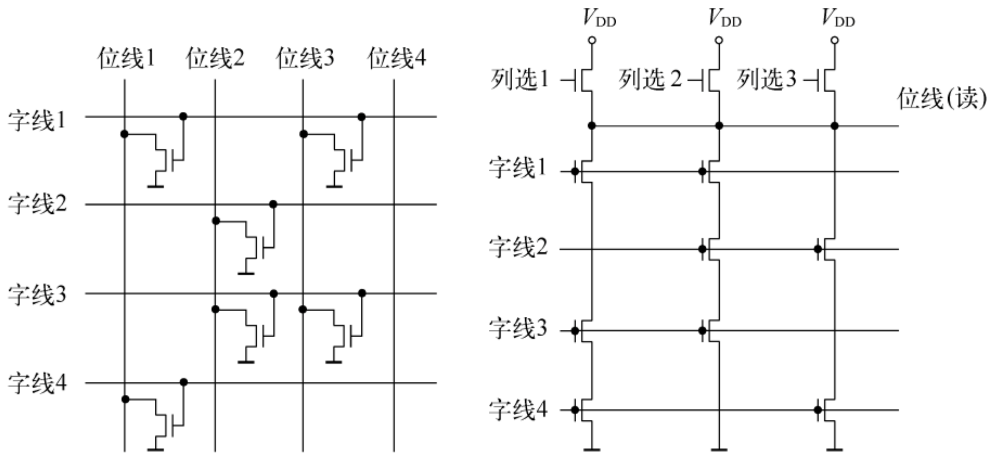


图 1　ROM 芯片分类



(a) NOR 阵列　　(b) NAND 阵列

图 2　掩膜 ROM 阵列

个 4 字 ×4 位掩膜 ROM 阵列，阵列中连接 MOS 管的数据为 1，反之为 0。图中存储的 4 个字分别为 1010，0100，0110 和 1000。这种 MOS 结构从字线方向看，是一个 4 输入端“或非”门，通常称为 NOR 阵列。读出操作时，经地址译码后在被选的字线上加高电平，非选字线为低电平。高电平字线使选中的 MOS 管通导，相应位线输出低电平，没有选中的 MOS 管的位线输出高电平。位线输出经反相后得到 ROM 存储的数据代码。除 NOR 阵列外，还有 NAND 阵列，如图 2(b)所示。

早期的掩膜 ROM 主要用于大型计算机操作系统存储媒体，20 世纪 80 年代中期电子游戏机的发展，特别是它对大容量、高集成度、低成本 ROM 的需求促成了掩膜 ROM 市场的发展，并且长盛不衰。据 1988 年统计，掩膜 ROM 市场的 54% 用于消费性产品，其中游戏机占 46%，电子乐器占 8%；而数据处理市场仅占 43%，主要用于文字处理机(19%)、个人计算机和工作站(10%)、办公室自动化(5%)、打印机等(9%)；工业应用市场很小，不及 3%。图 3 为 1 Mb CMOS 掩膜 ROM 芯片框图，与早期低集成度(如 1 kb ROM)产品相比，主要是增加了输入地址和输出锁存功能。

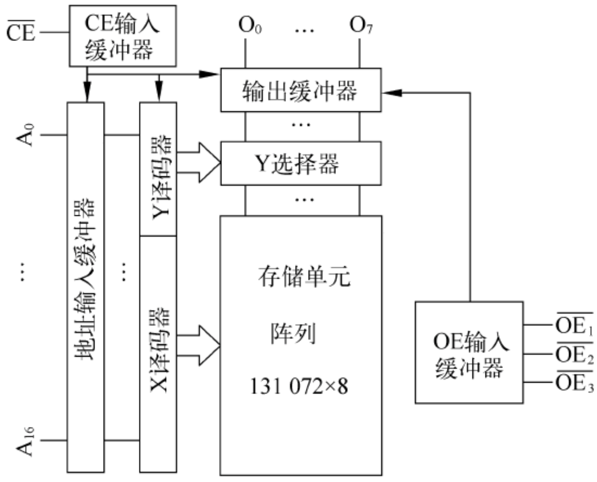


图 3　1 Mb(128 kb ×8)掩膜 ROM 芯片框图

可编程只读存储器芯片(PROM 芯片) 一种可按用户要求,由用户一次性写入内容的只读存储器芯片(参见**可编程只读存储器芯片**)。

可擦编程只读存储器芯片(EPROM 芯片) 一种可改写的只读存储器芯片,存储内容可由用户写入,芯片从电路系统中取出后,用紫外线照射可擦除存储内容(参见**可擦编程只读存储器芯片**)。

电可擦编程只读存储器芯片(EEPROM 芯片) 一种可改写的只读存储器芯片,其存储内容可由用户通过电信号进行写入和擦除(参见**电可擦编程只读存储器芯片**)。

快可擦编程只读存储器芯片(Flash EPROM 芯片) 一种可改写的只读存储器芯片,其存储内容可由用户通过电信号进行写入和擦除。与电可擦编程只读存储器芯片不同的是,其存储单元是单管结构(参见**快可擦编程只读存储器芯片**)。

非易失性随机存取存储器芯片(NV RAM 芯片) NV RAM 芯片是 ROM 芯片族中一个新系列,它将标准的 RAM 阵列和 EEPROM 阵列组合在同一芯片上。当芯片加电时,EEPROM 内容自动写入 RAM,提供原先保存的系统配置参数。系统运行中上述参数可通过 RAM 进行修改,并通过芯片内 RAM 和 EEPROM 之间的数据传送实现 EEPROM 芯片的重新编程,以实现系统配置参数的修改。NV RAM 芯片有两种形式:基于静态 RAM 单元阵列的 NV SRAM 芯片和基于动态 RAM 存储阵列的 NV DRAM 芯片。NV RAM 芯片具有随机存取和非易失的性能,适合于存储容量不大、存储参数不多、需高速存取并且仅仅偶然修改存储内容的场合。目前主要用于系统加电或断电时恢复和保存系统设定值,还广泛用于办公室可视终端、销售点终端、工业控制和机器人等。

NV RAM 芯片源于装有后援电池的 SRAM 芯片,即 B SRAM 组件。B SRAM 组件内存放一些锂电池,当外部电源切断时,组件内的电压监测电路控制将其切换到由锂电池供电,以保护 SRAM 内容,不使丢失。第一个 NV SRAM 芯片是 1977 年研制的,取其结构特征而称之为影像 SRAM。原型是一个 256 b 的 SRAM 芯片阵列,其上面像影子一样一一对应地覆盖 256b EEPROM。图 4 为 1988 年开发的 64 Kb NV SRAM 芯片产品的功能框图。图中的存储和陈列调用引脚表示了它独具的特性,可控制 SRAM 存储阵列和非易失 EEPROM 存储阵列之间的数据传送。该芯片内包含一个电荷泵,所以只需 5 V 供电。NV DRAM 芯片开发于 20 世纪 80 年代末,它保持了 DRAM 芯片的单管加一个存储电容的单元结构,只是该电容具有 EEPROM 芯片单元电容的非易失性能。

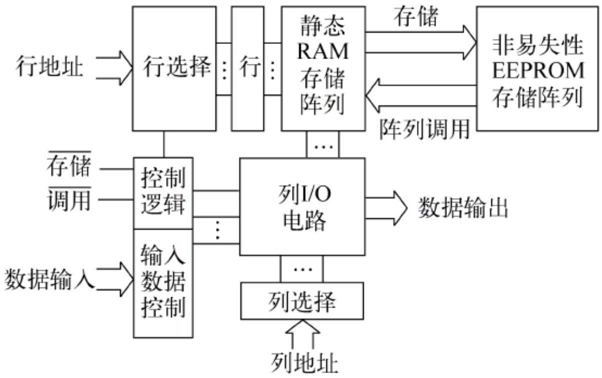


图 4 64 Kb NV SRAM 芯片功能框图

ROM 芯片的发展 随着人们对更高密度、更大带宽、更低功耗、更短延迟时间、更低成本和更高可靠性存储器芯片的追求和各种便携式产品的发展,在终端产品轻、薄、短、小的要求下,一些新型 NV RAM 技术相继出现,如铁电随机存取存储器、磁电阻式随机存取存储器和相变存储器等(参见**非易失性随机存储器**)。

半导体只读存储器芯片尽管灵活性差,但因具有高集成和低成本的特点,得以持续地占有非易失存储市场的很大份额。在半导体只读存储器芯片市场中,掩膜 ROM 芯片将继续占领一定的市场份额;现场可修改 ROM 芯片中,Flash EPROM 芯片和 NV RAM 芯片(特别是其中的 NV DRAM 芯片)由于功能特性好、成本低,将获得较快发展。

参考文献

1. Prince B. Semiconductor memories. 2nd ed. Chichester: John Wiley & Sons, 1991

2. Ma T P, Han J P. Why is nonvolatile ferroelectric memory field-effect transistor still elusive? IEEE Electron Device letters, 2002, 23: 386-388

3. Geppert L. The new indelible memories. IEEE Spectrum, 2003, 40(3): 48-54

(时万春)

zhidu guangdie qudongqi

只读光碟驱动器(read only optical disc drive) 只能从存储信息的光碟上读出数据的设备。只读光碟驱动器简称为只读光驱,主要为计算机所用。广义的只读光碟驱动器可指各种光碟播放

器(机),如CD播放机、DVD播放机等。

只读光碟驱动器由读出激光头、光斑自动聚焦机构、寻轨跟踪机构、主轴转速控制机构和它们的相应控制电路以及读出、纠错、接口电路等部分组成。激光头由激光发生器和接收器及光学透镜组组成。在自动聚焦机构和寻轨跟踪机构的配合下,激光束被聚焦对准光轨上代表存储信息的凹坑和平台并反射到光敏元件接收器上,读出电路处理接收器检测出的信号还原为数字信号,经纠错、接口电路处理传往计算机。在读出的过程中,光头与光碟表面不接触。

光碟驱动器标准规定1 X读出速度是150 KB/s,目前只读光碟驱动器最高读出速度达到56 X倍速。

只读光碟驱动器的平均读取时间(average access time)定义为每次从开始光头定位到开始读数据这个过程所需时间的平均值,它也是衡量光碟驱动器性能的一个重要参数。平均读取时间从300 ms现已缩短到75 ms。

只读光碟驱动器有三种不同的读取方式:①恒定角速度方式CAV(constant angular velocity),其马达转速恒定但在读不同直径光轨时传输速率有变化;②恒定线速度方式CLV(constant linear velocity),马达转速改变以求线速度恒定,即数据传输率保持恒定;③局部恒定角速度方式PCAV(partial constant angular velocity)。在CLV和CAV之间择优切换。

只读光碟驱动器采用了功能强大的错误码检测和纠正措施,基本对策归纳起来有三种:

(1) 采用CRC码(cyclic redundancy code)检测错误;

(2) 采用里德-索洛蒙码,简称为RS码,进行纠错;

(3) 交叉里德-索洛蒙码CIRC(cross interleaved Reed-Solomon code)进一步纠错。

只读光碟驱动器的计算机接口类型主要有IDE、EIDE、USB、SCSI、SCSI-2等(参见**外存储设备接口**),后两种接口的传输速度较快。

为了提高光碟驱动器机械性能,各个不同的生产厂家也推出了相应改善的技术:安装悬浮式减震橡胶;采用悬挂技术和橡胶减震支架;采用悬浮承载技术;采用先进的双重动态悬挂系统等。

只读光碟能存储大量数据、数字化的图像、视频、音频和多媒体信息,其存储容量由存储面积(盘片直径,单/双面,单/双/多层)和存储密度的乘积来决定。存储密度取决于激光波长和激光光斑直径。

小型光碟只读存储器CD-ROM(compact disc-read only memory)是1988年ISO/IEC 10149协议提出的、最早普遍采用的只读光碟,该协议规定CD-ROM盘片的直径是120 mm(4.75 in),厚1.2 mm。该光碟用折射率为1.5的透明材料(常用的是聚碳酸酯)制成。在存储媒体表面用母碟注塑工艺形成一连串的“凹坑”和“平台”,凹坑和平台在存储表面上组成一个螺旋线状的光轨。光轨的间距为1.6 μm,相当于每英寸16 000条光轨。凹坑的宽度约0.11 μm,深度为0.5 μm。采用780 nm波长激光读取。每片碟只使用一面,可存储数据容量为680 MB。1988年ISO 9660定义了CD-ROM上文件和目录的格式,此标准允许有不同操作系统的不同计算机访问同样的数据格式。通过ISO 9660之类的标准完成了媒体的全世界认同和彼此协作性。

数字多用途光碟只读存储器DVD-ROM(digital versatile disc-read only memory)是**数字多用途光碟**(digital versatile disc)原始五种规格中用于只读存储的一种规格。DVD采用650 nm波长的红光读取。120 mm单层DVD-ROM可存储数据容量为4.7 GB,双面双层可达17 GB。

DVD只读光碟驱动器是一种可以读取DVD-ROM光碟的驱动器,除了兼容读取DVD-ROM,DVD-AUDIO,DVD-VIDEO,DVD-R,CD-ROM等常见的格式外,对于CD-R/RW,CD-I,VIDEO-CD,CD-G等都要能很好的支持。

蓝光光碟只读存储器BD-ROM(blu-ray disc read only memory)是由SONY及松下电器等企业组成的“蓝光光碟联盟”策划的DVD-ROM的下一代光盘只读存储器规格,并以SONY为首于2006年开始全面推动相关产品。采用波长405 nm的蓝色激光光束来进行读操作,一片单层的蓝光只读光碟的容量为25 GB或27 GB。双层可达到46 GB或54 GB。而4层及8层蓝光只读光碟容量分别为100 GB和200 GB。

蓝光只读光碟驱动器是一种可以读取BD-ROM蓝光光碟的驱动器。亦兼容读DVD-ROM,CD-ROM。

只读光碟驱动器技术发展中催生了既可只读又可刻录的光碟驱动器产品,一类是COMBO光碟驱动器,它集合CD刻录、CD-ROM和DVD-ROM功能为一体。另一类包括了CD-R、CD-RW、DVD、蓝光刻录机

等,其中 DVD 刻录机又可分别兼容 DVD + R、DVD-R、DVD + RW、DVD-RW(W 代表可反复擦写)和 DVD-RAM 光碟;蓝光刻录机有 BD-R(单次烧录)及 BD-RE(多次烧录)方式。

参考文献

Pohlmann K C. The compact disc: A handbook of theory and use. Oxford: Oxford University Press,1993

(余胜生)

zhicheng yuyi

指称语义(denotational semantics)　把语言成分映射为数学对象,然后用定义在对象上的运算所表达出的语言的语义。

指称语义的主要思想是英国牛津大学 C. Strachey 于 1964 年前后提出的。美国 D. Scott 创建的论域理论为指称语义学奠定了数学基础。论域理论是讨论各种语言成分指称物的数学结构,以及提供在这种数学结构上定义语言成分的语义和推导语言成分特性所必须的数学工具。

IBM 公司维也纳实验室在研究操作语义方法 VDL 的基础上,于 20 世纪 70 年代初转向指称语义的研究,发展了基于指称语义方法的一整套开发软件的工程方法,称为维也纳开发方法(简称 VDM)。

1976 年英国一些学者发展了论域理论,提出幂论域理论,从而为定义非确定性程序的指称语义奠定了理论基础。指称语义方法定义语言的语义基本思想是:先确定指称物,然后给出语言成分至指称物的语义映象。这个映象要求满足:

(1) 每个成分都对应有指称物;

(2) 复合成分的指称只依赖于它的子成分的指称。

下面以算术表达式和赋值语句及顺序语句为例给以简要说明。

算术表达式的效果就是根据程序变量当前值计算表达式的值。程序变量的当前值可以用一个数值向量来表示。如果有 k 个程序变量 $x_1,x_2,\cdots,x_k$,则 k 维数值向量$(n_1,n_2,\cdots,n_k)$表示 x_1 的值为 n_1,x_2 的值为 n_2,……,x_k 的值为 n_k。程序变量的一种取值称为程序的一个状态。状态的全体集合称为状态空间,记作 State。算术表达式的值的范围记作 Num。算术表达式的指称物就是 State 至 Num 的一个映象,也就是根据 State 中的任意一个元素(即程序变量的一组取值)可求得 Num 中对应的一个元素(即表达式在变量的这组取值下的值)。数学中的映象只反映集合间元素的对应,用映象作为语言成分的指称物在语义的定义中就避免了涉及语言成分的执行过程。State 至 Num 的全体映象,即全体表达式的指称物,记作 State →Num。不同表达式的指称物是不同的,对算术表达式的语义下定义,就是要对每个算术表达式至其指称物的一个对应下定义,故也可表示为一种映象。用 Exp 表示算术表达式的全体,那么 Exp 的语义,就是 Exp 至指称物集合(State, Num)的映象,记作

$$\text{Exp} \to (\text{State} \to \text{Num})$$

常量 n 是一种算术表达式,变量 x 本身也是一种算术表达式,两个算术表达式的和及积等也是算术表达式,故算术表达式 Exp 的抽象语法可用巴科斯范式(BNF)表示:

$$E ::= n \mid x \mid e+e \mid e\times e \mid \cdots$$

指称语义映象 D 可定义如下:

$D[n](S)=n$

$D[x_i]=S_i$

$D[e_1+e_2](S)=D[e_1](S)+D[e_2](S)$

$D[e_1\times e_2](S)=D[e_1](S)\times D[e_2](S)$

第一个公式表示,无论程序处于何种状态 S,常量 n 的值不变,为数学中相应的量 n。为了区分元语言和程序设计语言,指称语义定义中将程序设计语言中的成分用[及]括起来。

第二个公式表示,变量 x_1 在状态 S 时的值为数值向量 S 的第 i 个分量 S_i。

第三、四个公式表示,表达式(e_1+e_2)和 $e_1\times e_2$ 在状态 S 时的值分别为子表达式 e_1 和 e_2 在状态 S 时的值的和与积。

表达式的抽象语法规定如何用最简单的表达式常量和变量构成其他表达式。而表达式的语义定义也是先给出最简单的表达式的语义,然后按照语法的构造过程去定义其他表达式的语义,使得复合成分的语义由各成分语义复合而成。这种定义语义的方法叫作结构式方法,或叫语法引导方法。指称语义学就是一种结构式形式语义学。执行程序设计语言中的语句导致程序状态的改变(即程序变量取值的改变),故语句的指称物应是 State 至 State 的映象:State →State。定义语句的语义就相当于定义映象 D:

$$D: \text{Sts} \to (\text{State} \to \text{State})$$

Sts 表示所有语句的集合。如表示赋值语句$(x_i := e)$的语义即改变状态 S 为 $S\Big|_{i}^{D[e](S)}$。

$S\Big|_{i}^{D[e](S)}$ 表示将 S 中的第 i 个分量(即 x_i 的值)改变为表达式 e 在状态 S 时所取的值(即 $D[e](S)$)。

$$D[S_1;S_2](S) = D[S_2](D[S_1](S))$$

表示顺序执行语句 S_1 和 S_2,即先执行 S_1,并将执行 S_1 后形成的新状态,作为当前状态再执行 S_2。

实际的程序设计语言非常复杂,所定义的语义映象比之上文列举的远为复杂。为了允许同名变量在不同过程体中表示的值不同,指称语义中引进环境的概念;为了刻画程序控制的转移又引进后续的概念,这些概念在描述和简化语义定义中有重要的作用。

参考文献

周巢尘. 形式语义学引论. 长沙: 湖南科技出版社,1985

(周巢尘　李晓山)

zhiling geshi

指令格式(instruction format)　计算机指令的表示形式。

一条指令通常包含下列信息:

(1) 操作码　说明指令所执行的操作,例如加法、移位或转移等。一台计算机可能有几十条甚至几百条指令,每一条指令都有一个相应的**操作码**。操作码的长度与机器所含的指令条数有关。

(2) 操作数来源　在指令中指出操作数的地址(寄存器地址或存储器地址),根据地址取得操作数。

(3) 目的地址　运算结果保存在目的地址中,可以是寄存器地址或存储器地址。

(4) 下一条指令的地址　该地址由**程序计数器**提供。当程序顺序执行时,每执行一条指令后,程序计数器的内容加 1,得到下一条指令的地址,而当执行转移指令(或其他改变执行顺序的指令)时,由本条指令指出下一条指令的地址。

指令由操作码和地址码组成,地址码可包括上述(2),(3),(4)中的部分内容。根据地址码部分所包含的地址个数来划分指令,可以有零地址、一地址、二地址和三地址等多种指令格式,如图 1 所示。其中 OP 为操作码,A_1,A_2,A_3 为**地址码**。

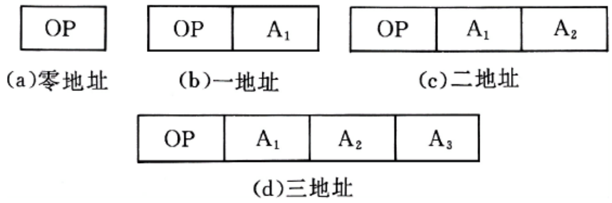


图 1　指令格式

零地址指令只有操作码,没有地址码,适用于两种情况:①空操作、停机等不需要地址码的指令;②操作数地址是隐含的,如堆栈指令,由堆栈指针给出地址。

一地址指令在指令中只给出 1 个地址,例如"加 1"、"减 1"等单操作数指令就采用这种格式。又如某些计算机的双操作数指令,已约定一个操作数在某个指定的寄存器中,指令中只需要指出另一个操作数地址,运算结果通常也是送回到这个约定的寄存器中。

二地址指令是广泛被采用的双操作数指令,A_1,A_2 是两个操作数的地址,其中一个(A_1 或 A_2)还是目的地址,算术逻辑运算指令一般都是二地址指令。

某些计算机采用三地址指令,其中两个地址为两个操作数的地址,第三个为目的地址。

某些指令直接给出操作数(参见**寻址方式**),转移类指令在地址码给出程序的转移地址。

以上所述是经常见到的几种指令格式,一台计算机可能只具有其中一部分指令格式。寄存器地址的长度只需要几位就够了,存储器地址的长度取决于寻址方式和存储器容量,一般在 20 位到 32 位之间。

精简指令集计算机的指令长度一般是固定的,操作码和地址码的长度及其在指令中的位置一般也是固定的。**复杂指令集计算机**的指令长度一般不是唯一的,它随指令而异,有的计算机的指令长度甚至在一个字节到十几个字节之间变化,操作码的长度也随之而变。应尽量把常用指令的长度设计得短一些,以节省程序的存储空间和减少访问存储器的次数。

(王爱英)

zhilingji bingxing chuli

指令级并行处理(instruction level parallel processing, ILPP)　在处理机的指令级上提高指令执行并行性的软件和硬件相结合的技术。

指令级并行处理是在**精简指令集计算机**(RISC)问世以后发展起来的,其目标是提高处理机本身的处理速度,而对于用户来说是完全透明的。

先进的**处理机体系结构**都采用流水线技术(参见**计算机流水线**),使多条指令的执行重叠进行,以

提高处理机的运行速度。流水线执行中存在转移相关和数据相关问题。为了解决这些问题,在软件方面,采用编译优化,尤其着重其中的指令调度重组;在硬件方面,有解决数据相关的相关检测电路和内部提前数据通路等技术以及解决转移相关的转移预测技术等。

在**多发射结构**中,每个机器周期发射多条指令,不仅同一个机器周期内的多条指令之间存在着相关性,而且邻近周期的各条指令之间也存在着同样的问题。因此,解决流水线中的相关性问题更为困难。

解决流水线中相关性问题可以采用硬件和软件相结合的指令调度技术。通过指令调度以解除或减少程序中指令间的相关性,使更多的指令可以并行执行而保持流水线畅通,从而提高处理机执行指令的速度。

衡量指令执行并行性的参数为 IPL,即

IPL = 程序执行总指令数/程序执行总周期数

进行指令调度的目的是提高 IPL 值。

指令调度应在一个确定的范围内进行,通常称这个范围为一个基本块。基本块的定义是:它是只有 1 个入口点和 1 个出口点的代码段。该代码段中没有转移或跳转等分支指令。段头为入口点,段尾为出口点。在一个基本块内进行指令调度,避免了转移相关问题,只需解决数据相关和资源冲突问题。但是,经过统计分析,一般程序中的基本块平均包含的指令数只有 6 ~ 8 条。这使指令调度的作用有限,IPL 一般不超过 2。

要进一步提高 IPL,必须考虑跨基本块之间的指令调度,或者扩大基本块。如把从指令存储器取出的、放在指令缓冲器中的顺序指令流,经过一定的调度以后,乱序执行,可以扩大基本块。另外,循环在程序中是频繁出现的,如果把循环优化并使各个循环中的指令得以重叠执行,也可使 IPL 提高较多。这些都是指令级并行处理的内容。

基本块以内的指令调度称为局部调度;跨基本块的指令调度称为全局调度。从程序执行角度,调度可以分成静态调度和动态调度。静态调度是先把源程序进行编译,生成目标码,再进行调度,然后执行;动态调度是在执行目标码的过程中进行调度。前者主要靠软件方法实现,后者则用软件和硬件相结合的方法实现。

从硬件实现调度的角度,也可有两种级别上的调度:一是在处理机组成的级别上,如转移预测缓冲、寄存器记分牌和内部提前技术;二是在处理机体系结构的级别上,如多发射结构。

基本块以内的指令调度方法有寄存器分配法、指令重新排序法、指令压缩法等,其中所使用的算法有线性压缩算法、搜索算法、列表调度算法和关键路径算法等。

跨基本块的指令调度方法有静态转移预测法、动态转移预测法、踪迹调度法和冒险执行法等。

指令级并行处理对处理机体系结构和编译优化技术有重大影响,对提高处理机的性能起关键性作用。

参考文献

李三立,李亚民. RISC——单发射与多发射体系结构. 北京:清华大学出版社,1994　（李三立）

zhiling jicunqi

指令寄存器(instruction register)　用来存放当前正在执行的指令的寄存器。计算机的所有操作都是通过分析存放在指令寄存器中的指令后再执行的。当指令从主存储器取出后,将其暂放在指令寄存器中,以便在指令执行过程中,完成一条指令的全部功能控制。指令内容包含有确定操作类型的操作码和指出操作数来源或去向的地址。操作码部分送到译码电路进行分析,指出本指令将执行何种类型的操作;地址部分送到地址形成部件生成有效地址,作为取数或存数的地址。　（刘恩德）

zhiling leixing

指令类型(instruction type)　根据指令完成的主要功能而对指令所进行的分类。

一台计算机通常有几十条到几百条指令,按其所完成的功能可分为:算术逻辑运算指令、移位指令、浮点运算指令、十进制运算指令、数据传送指令、转移指令、字符串处理指令、向量运算指令、堆栈指令、输入输出指令、特权指令、控制指令和多处理机指令等。但并不要求一台计算机具有上述所有类型的指令。

(1) 算术逻辑运算指令　计算机一般都具有这类指令。早期的小型计算机和微型计算机的硬件结构比较简单,一般只设置二进制定点加减法、比较和求补码(取负数)等最基本的算术指令。由于芯片集成度的提高,后来的中央处理器都支持用硬件实现的乘除法指令。计算机还具有对两个操作数进行逻辑乘、逻辑加和按位加(异或)等操作的逻辑运

算。有些计算机还设置位操作指令，如位测试（测试指定位的值）、位清除（把数中的某一位设置为零）、位求反（取某位的非值）等指令。

（2）移位指令　分算术移位、逻辑移位和循环移位3种。可以对 n 位操作数左移或右移 $0\sim n-1$ 位。算术移位和逻辑移位的主要差别在于右移时，填入高位的数码不同，算术移位处理的是带符号的操作数，所以右移时保持最高位（符号位）不变；逻辑移位处理的是不带符号的操作数，右移时，最高位填入0。算术左移和逻辑左移的操作是相同的，低位补充0。循环左移将移出的最高位送到最低位；循环右移将移出的最低位送到最高位，使数据本身循环传送。

移位还可实现对带符号数或不带符号数乘以 2^i 或整除以 2^i 的运算（分别左移 i 位或右移 i 位）。移位指令的执行时间比乘除法运算的执行时间短得多。

（3）浮点运算指令　浮点运算适合对数值范围变化较大的数，用于科学计算或工程计算的计算机往往设置浮点运算指令，处理的数经常为单精度（32位）或双精度（64位）数。某些计算机没有设置浮点运算指令而用子程序实现浮点运算，因而处理速度慢。

（4）十进制运算指令　在某些数据处理系统中，输入输出数据很多，设置十进制运算指令可减少十进制数与二进制数之间的转换，提高处理速度。在通用大型计算机系统中，这些数由若干位十进制数码组成，每个十进制位用BCD码表示；在某些微处理器中设置的十进制加减法运算指令往往只对1位十进制数进行运算，对于多位的十进制数的运算则用程序实现。

（5）数据传送指令　这类指令实现寄存器与寄存器、寄存器与存储器单元、存储器单元与存储器单元之间的数据传送或数据交换。某些计算机I/O设备中的寄存器与存储器单元统一编址，此时的数据传送指令还包括通用寄存器（或存储器单元）与I/O设备寄存器之间的数据传送。

数据传送指令1次可传送1个数据或1批数据。传送1批数据时，执行时间较长，在传送过程中如果有中断请求，往往允许暂时中止现行指令的执行而转入中断处理程序，处理完毕返回原程序后，再从该指令中止处继续执行下去。

（6）转移指令　这类指令用于控制程序流的转移。在大多数情况下，计算机是顺序执行程序的，但也会遇到转移到另一段程序或循环执行某段程序的情况。

按转移的性质，转移指令分为无条件转移、条件转移、过程调用与返回、软中断等几种。①无条件转移指令　不受任何条件约束，将程序转移到本指令所规定的下一条指令继续执行。②条件转移指令　根据指令所指定的条件进行测试，根据测试结果的"真"、"假"来决定转移是否执行。通常利用算术运算指令生成的条件码来控制程序流，实现程序的分支。③调用指令和返回指令　在编写程序过程中，常常预先编写一些经常使用的，能够独立完成某一指定功能的程序段，在需要时能随时调用，而不必多次重复编写，以节省存储器空间和简化程序设计，这些程序段称为子程序或过程。操作系统也提供大量通用子程序，如申请资源、读写文件、控制外部设备等，需要时可直接调用。计算机使用调用指令Call（过程调用、系统调用、转子程序）来实现从一个程序转移到另一程序段的操作，执行完被调用的程序段后再返回到原调用程序，继续执行，这一功能由返回指令Return实现。返回地址一般保留在堆栈中，随同保留在堆栈中的还有一些状态寄存器和通用寄存器的内容。④**软中断**（或称**陷阱**）和软中断指令　在计算机运行过程中，有时可能出现电源电压不稳定、存储器校验出错、输入输出设备出现故障、用户使用了未定义的指令或特权指令等意外情况，使计算机不能正常工作，如果不及时处理，会影响系统正常运行。此时，计算机发出软中断信号，并暂停当前程序，转入故障处理程序。在某些计算机中设置有可供用户使用的软中断指令，利用它来实现系统调用和程序请求。

（7）字符串处理指令　用于信息管理、数据处理、办公室自动化等领域的计算机需要有很强的非数值处理能力。字符串处理指令是一种非数值处理指令。它一般包括字符串传送、字符串比较、字符串查询、字符串转换等指令。字符串传送指令可将数据块从主存的某区域传送到另一区域；字符串比较指令将一个字符串与另一字符串逐个字符进行比较，以确定是否相等；字符串查询指令可查找在字符串中是否含有某一指定的字符或子字符串；字符串转换指令可将字符串从一种表达形式转换到另一种表达形式，例如从ASCII码转换成EBCDIC码（扩充的二-十进制交换码）。

这类指令可用于需对大量字符串进行处理的文字编辑和自动印刷排版系统中。

（8）向量运算指令　一般在大型计算机或巨型计算机中设置这类指令，可对向量或矩阵进行求和、求积等运算。

（9）堆栈指令　**堆栈**是由若干个连续的存储单元组成的先进后出存储区，第一个送入堆栈的数据存放在栈底，最后送入堆栈的数据存放在栈顶。栈底是固定不变的，而栈顶却随着数据的进栈和出栈不断变化，有一个专用的寄存器或指定的存储器单元用来存放栈顶地址，这个寄存器或存储器单元称为**堆栈指针** SP。由于堆栈具有先进后出的性质，因而在中断、子程序调用过程中广泛用于保存返回地址、状态标志和现场信息等。

在具有堆栈结构的计算机中，堆栈是提供操作数和保存运算结果的主要存储区，包括算术逻辑运算指令的大多数指令通过访问堆栈获得所需的操作数和保存操作结果。在一般计算机中有压入（堆栈）和弹出（堆栈）两种指令。压入指令（PUSH）把指定的操作数送入堆栈的栈顶，弹出指令（POP）把栈顶的数据送到指令所指定的目的地，并相应地修改堆栈指针。

（10）输入输出指令　计算机所处理的一切原始数据和所执行的程序（除了固化在只读存储器 ROM 中的以外），均来自输入设备，处理结果需通过输出设备输出，这些通常是由输入输出指令完成的。某些计算机采用输入输出设备和存储器统一编址的方法把输入输出设备中的寄存器看成是存储器的某些单元，任何访问存储器的指令均可访问输入输出设备，因此不再设置输入输出指令。

（11）特权指令　某些指令使用不当会破坏系统或其他用户信息，为了安全起见，这些指令只能用在操作系统或其他系统软件中，而不提供给用户使用，称为特权指令。

一般来说，在单用户、单任务计算机中不一定需要特权指令，而在多用户、多任务计算机系统中，特权指令是必不可少的。它主要用于系统资源的分配和管理，包括改变系统的工作方式，检测用户的访问权限，修改虚拟存储器的段表、页表和完成任务的创建和切换等。

在某些多用户计算机系统中，为了统一管理所有的输入输出设备，把输入输出指令也作为特权指令，不允许用户直接使用，当程序需要输入输出时，通过陷阱指令进入操作系统，由操作系统控制输入输出操作。

（12）控制指令　包括等待指令、停机指令、空操作指令、开中断指令、关中断指令、置条件码指令、置程序状态字指令等。

当用户程序执行完毕后，可安排一条停机指令。但在多用户情况下，不允许停机，因为其他用户程序可能正在等待使用机器，此时可安排一条等待指令或执行只有少量指令的小循环程序（动态停机）。

空操作指令除了将程序计数器增量外，不进行其他操作。

开中断指令、关中断指令一般是进入中断处理程序后用的。

（13）多处理机指令　在多处理机或多处理器系统中，为了管理共享的公共资源和相互通信，一般设置“测试与设定”或“数据交换”指令，这些指令最主要的特点是在指令执行过程中不允许打断，用以防止多个处理器同时读取和修改共享数据区。

以上涉及的是在计算中通常遇到的指令类型，在每一类型中还可能包括有数量不等、繁简各异的指令，同一条指令还可能有多种寻址方式。随着集成电路工艺的改进和集成度的提高，计算机应用范围的不断扩大，计算机的体系结构不断发展，指令系统也会发生相应的变化。（王爱英）

zhiling xitong

指令系统（instruction set）　一台计算机中的所有指令的集合。又称指令集。程序员用各种语言编写的程序要翻译（编译或解释）成以指令形式表示的机器语言以后，才能在计算机上运行。**指令**由规定计算机操作类型及操作数地址的一组字符组成，在计算机内部用二进制码表示。计算机硬件完成各条指令所规定的操作，并保证按程序所规定的顺序执行指令，所以指令系统反映了计算机的基本功能，是硬件设计人员和程序员都能见到的机器的主要属性。

早期，由于组成计算机的元器件价格较贵，为了降低成本，硬件尽量简单，那时计算机的指令系统是很简单的，一些比较复杂的运算或操作由子程序实现，因此计算机的处理速度很慢。

随着元器件的性能提高和价格下降，硬件成本在一个计算机系统中所占的比例不断下降，而软件成本不断上升。为了便于编程和降低软件成本，计算机的指令系统不断扩充，指令功能逐步增强。指令系统的改进围绕着缩小指令与高级语言语句的语义差异和有利于提高操作系统的执行效率而进行。例如高级语言中的实数计算可通过浮点运算指令实

现;某些应用需对数组进行运算时可通过向量指令实现;为了便于程序嵌套,设置了调用指令和返回指令;为了便于操作系统的实现和管理,设置了控制系统状态的特权指令、管理多道程序和多处理机系统的专用指令等。

美国 IBM 公司在 1964 年推出 IBM 360 计算机时宣布了系列计算机的思想。从此以后,各计算机公司生产各自的系列计算机。系列计算机的特征是指令系统、数据格式、I/O 接口等保持相同,因而软件兼容。即使新型计算机或高档计算机扩充了指令系统,但仍保持软件向上兼容的特点,因此保护了用户在软件上的投资。

但是日趋庞大的指令系统不但使计算机的研制周期变得很长,实现起来也很复杂,有时还会降低系统性能。1975 年 IBM 公司提出精简指令集的想法,后来美国加州伯克利大学的 RISC Ⅰ机和 RISC Ⅱ机、斯坦福大学的 MIPS 机研究成功,对**精简指令集计算机**的发展起了很大作用。此后将过去的指令系统比较庞大的计算机称之为**复杂指令集计算机**。目前这两个类型计算机都存在,并在技术上相互推进。

参考文献

1. 王爱英. 计算机组成与结构. 4 版. 北京:清华大学出版社,2007

2. 孙强南,孙昱东. 计算机系统结构. 北京:科学出版社,1992

3. 李学干,苏东庄. 计算机系统结构. 西安:西安电子科技大学出版社,1991

(王爱英)

zhiling zhouqi

指令周期(instruction cycle) 计算机执行一条机器指令的过程与时间。

计算机执行一条指令的过程可以划分成若干阶段,每个阶段完成一定的功能。这种具有特定功能的时间段称为**机器周期**,也叫 CPU 周期。

一条典型指令的执行过程如下:第一步是取指令,从指令计数器中把要执行的指令的地址送入内存地址寄存器,从存储器中取出指令,送到指令寄存器;第二步是分析指令,对指令寄存器中的操作码进行译码,确定指令应该完成的功能,并对寻址方式进行解释,确定如何寻找操作数的有效地址;第三步是取数,根据操作数的有效地址,取出操作数;第四步是执行,按照操作码的规定,对操作数进行具体处理;第五步是回送处理结果。上述具有特定功能的各步时间段分别叫作取指令周期、分析指令周期、取操作数周期、执行周期和回送结果周期。不同指令需要不同的机器周期,例如有的指令寻址方式是变址寻址或间接寻址,为了求出操作数的有效地址,在指令周期中还需增加变址周期或间址周期。为了提高输入输出(I/O)数据传送速度,还需要设置中断周期、直接存储器存储(DMA)周期等。但最短的指令必须包含两个机器周期:取指周期和执行周期。

机器周期的时间宽度取决于完成该周期的功能所需的时间。不同机器周期所需的时间是不同的。为了设计简便,在划分指令功能阶段时,使完成每个阶段所需的时间差不多。具体设计时,使各个机器周期宽度相同。因此机器周期的宽度应取决于各周期中所需时间的最长者。每个机器周期内各种操作的时间顺序由节拍电位来指定。典型的解决方案是将一个机器周期分为有一定时间顺序的若干节拍,将有关操作命令安排在不同时间节拍中,以实现对各种操作的顺序控制要求。以取指令周期为例:该周期的第 1 节拍将要执行的指令地址由指令计数器经地址总线送往内存地址寄存器,第 2 节拍向内存发读出命令,经过若干时间等待,最后一个节拍将读出的指令经数据总线送到指令寄存器中。节拍电位的宽度一般决定于中央处理器执行一次运算或寄存器间完成一次数据传送所需时间,通常与计算机的时钟周期一致。

不同的机器周期需要的节拍数目是不同的,因此机器周期的宽度也是不同的。为了设计实现方便,又能满足指令要求,可以采用以下几种方案决定机器周期的宽度:①统一节拍法,所有机器周期的节拍数都是一样的,取其中最大节拍数。这样控制比较简单,但对某些周期,时间有些浪费。②可变节拍法,允许各种机器周期的节拍数不同,但控制复杂一些。③延长节拍法,根据多数机器周期需要的节拍数,作为其基本节拍数,特定情况可延长一个节拍或两个节拍。④插入节拍法,当某个节拍的操作不能完成时,可随机插入若干个等待节拍。

设计高速计算机的主要目标是提高指令运算速度,这就要求压缩指令周期长度、减少每条指令中机器周期的数目、压缩机器周期的时间宽度,减少每个机器周期中节拍个数和节拍电位宽度,提高时钟频率等。在**精简指令集计算机**指令中,除了两条访存指令外,大都采用寄存器-寄存器型寻址,CPU 又设置大量通用寄存器,操作数直接由寄存器取出,减少访存取数周期,缩短取数时间,有利于压缩指令周期时间,提高指令速度。采用流水线技术,使多条指

令重叠并行执行，有效地提高计算机处理速度。例如 Intel Pentium 计算机的整数指令，有 5 个机器周期，即 PF 指令预取、D1 指令译码、D2 生成地址、EX 执行、WB 写回结果。指令流水线中对应采用 5 级流水线结构，每个流水级的操作都在一个时钟周期内完成。理想情况下，一条 5 级流水线可同时有 5 条指令重叠执行，在每一时刻，5 条指令分别处于不同的流水级。机器运行时，每一个时钟周期流水线读入一条指令，每一个时钟周期完成一条指令，得到一条指令的计算结果，计算机的处理速度提高 5 倍。每条指令平均占用 1 个时钟周期。如果采用超标量结构，每个时钟周期可同时从指令高速缓存中读出多条指令并行操作，机器速度将有更大提高，例如 DEC Alpha，Super SPARC，Intel 80860 都是采用超标量结构。超流水线结构的每一级流水线分解为多级更短的流水线，例如 MIPS R4000 将原来的 4 级流水线变成 8 级流水线，在一个基本时钟周期内分时发出两条指令，从而进一步压缩完成一条指令的时间。

参考文献

李三立，李亚民. RISC 单发射与多发射体系结构. 北京：清华大学出版社，1993

（谢树煜）

zhiwen shibie

指纹识别（fingerprint recognition）　计算机利用人手指上的条状纹路信息自动识别其身份的过程和技术。指纹是人手指表面的皮肤凹凸不平产生的纹路，手指皮肤的凸起部分称为脊，凹陷的部分称为谷。脊和谷的不同分布，形成了不同的指纹结构特征。

指纹作为生物特征的身份鉴别方式，具有稳定性和唯一性两大优点。首先，指纹具有很强的相对稳定性。指纹在人体胚胎发育的第 3～4 月开始生长，第 6 个月完全形成。指纹一旦形成，尽管随着年龄变化，在外形大小、纹线粗细上会产生一定的变化，而且局部纹线也可能出现新的特征，然而其纹线的类型、结构、统计特征的总体分布等，却是稳定的，没有明显的变化。此外，对于外界伤害的影响指纹也是相对稳定的。比如手指皮肤受伤，只要不伤及真皮层，伤愈以后的指纹纹线仍能恢复原状；如果伤及真皮层，伤愈以后的伤疤则形成新的稳定特征。其次，指纹具有唯一性。指纹的形成依赖于胚胎发育时的环境。每个人的指纹都是独一无二的，而且即使同一个人的十个手指的指纹也有明显的区别。从指纹的特征来看，每个指纹一般有 70～150 个基本特征点。从概率的角度，如果两枚指纹中有 12～13 个特征点吻合，则认为是同一指纹。而按照现有的人口计算，上述概率需要 120 年才可能出现两枚完全相同的指纹。

指纹识别技术利用指纹纹路的结构特征来进行分类识别。一般指纹纹路的结构特征可以分为全局结构的纹形特征和局部结构的细节特征。指纹识别时首先利用全局纹形特征进行粗分类，然后再利用局部细节特征进行匹配。

从指纹的全局结构来看，有如下一些纹形结构：①弓形，包括平弓形、帐弓形；②箕形，包括放射性箕形、尺骨状箕形；③斗形，包括平斗形、中心对称箕、双箕形等。图 1 举例说明了几种典型的纹形结构。



弓形　　箕形　　斗形　　双形

图 1　几种典型的纹形结构

指纹的局部结构特征主要是指指纹纹路的端点、分叉点等，称为细节特征，如图 2 所示的小桥、环、分叉点、三角点、端点等。

在识别时通常主要使用分叉点和端点两个主要的细节特征。其基本思想是首先提取指纹纹路上的分叉点和端点，并记录其位置和方向，然后按照最优

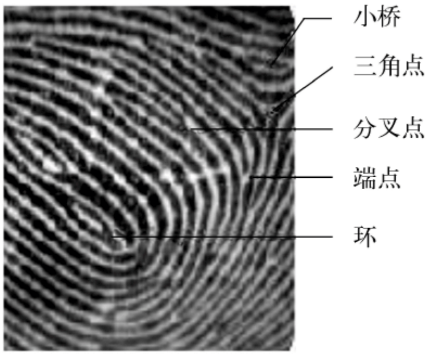


图 2　指纹的细节特征

准则,确定待识别指纹的特征数据相对于指纹模板的平移和旋转,然后将其对准,计算匹配上的特征点的个数、比例等参数,据此来判断待识别指纹是否与指纹模板匹配,从而确定识别结果,如图 3 所示。



图 3　指纹细节特征的提取
(左为指纹原图,右为提取的纹线和特征点图)

指纹识别作为个人身份识别最常用的技术之一,其算法已经比较成熟,并取得了较好的识别效果。随着人们对于安全性和鲁棒性的要求不断提高,现阶段的指纹识别在保证安全的前提下普遍存在错误拒绝率较高的问题。指纹识别的另外一个难点在于活体指纹的检测。如何解决上述问题,是今后需要进一步关注和研究的内容。

参考文献

1. http://www.sinobiometrics.com/

2. Jain A, Pankanti S. Automated Fingerprint Identification and Imaging systems. In: Advances in Fingerprint Technology (second edition). Elsevier, New York, 2001

3. Jain A, Hong L, Pankanti S. Biometrics: Promising Frontiers for Emerging Identification Market. Comm. ACM, 91-98, February, 2000

(蔡莲红　吴志勇)

zhizhen leixing

指针类型(pointer type)　由单一的空值及一组标识值组成的数据类型。又称指引元类型。每个标识值标识一个对象,被标识的对象的类型一般是确定的,也可以不确定(如 PL/1 语言)。被标识的对象和标识它们的值是在程序运行时创建和取消的。值 nil 是一个指针常量,它表示不指向任何对象(或说指向空)。它和任何指针类型都是赋值相容的。

PASCAL 语言中,指针类型定义为

类型名 = ↑对象类型

其中,对象类型可以是任何类型,在实际应用中一般为**记录类型**。两个指针类型一致,指的是它们的对象类型一致。PASCAL 语言允许在定义指针类型时,出现先引用后定义的情形。例如,下列类型定义

```
type pointcomp = ↑complex;
     complex = record
               re, in: real
               end;
```

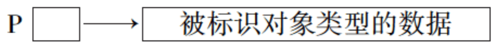
是正确的。

C 语言中指针类型定义为

```
type * name;
```

这里,type(类型)是 C 语言的任何一种有效类型(也叫基类型),name(名字)是指针变量的名称。指针的基类型定义了该指针所指向变量的类型。C 语言中有两个特殊的一元指针运算符:& 和 * 。& 的功能是回送其操作数的内存地址。* 与 & 相反,它回送的是存放在其后的地址中变量的值。

指针对象和**被标识对象**的关系可由下图表示:



在程序说明部分,只说明了指针对象 P,被 P 标识的对象是在程序运行中产生的,所产生的对象的值称为动态对象。动态对象是通过执行语言的预定义过程 new(如 PASCAL, Ada)创建的。其撤销或通过执行预定义过程 dispose(如 PASCAL)或当程序执行到离开其指针对象说明所在的作用域时自动撤销(如 Ada)。

指针类型的操作有相等及不等比较。在程序语言中,一般的动态数据(如栈、链表、队列、树等)是通过指针实现的。　　(陈涵生)

zhineng jiqiren

智能机器人(intelligent robot)　具有人类所

特有的某种智能行为的机器。它是一类具有高度自主性的自动化机器或设备，是机器人技术发展的高级形态。智能机器人技术涉及力学、机械学、电子学、控制论、计算机、**人工智能**和系统工程诸多领域的边缘学科。

在早期的科幻作品里，机器人是一种能听命于人、能从事劳动、形状像人甚至表现出人的某些思维和行为的人造机器。人类对这种先进生产工具的最初想象里，就要求它具有人类自身所特有的智能。

遥控操纵器和数控机床的出现为机器人的产生在技术上奠定了基础。用于危险或有害环境的遥控操纵器主要由两台利用传动机构连结的多关节机械手臂组成，人操作其中的主机械手，那么位于环境现场的从机械手就能跟随复现人手的动作，代替人完成作业。数控机床根据预先储存的工件切削模型以及刀具切削动作序列等数据，对机床的伺服轴进行运动控制，若要改变控制只需改变储存的数据。

1954 年美国人 George. C. Devol 设计并制作了世界上第一台机器人实验装置，发表了《适用于重复作业的通用性工业机器人》一文，并获得了专利。这台机器人把遥控操纵器的多自由度关节型连杆机构与数控机床的伺服轴连结在一起，编程输入预定的机械手动作后，就可以离开人的辅助而独立运行。此外，操作人员还可以用手带动机械手依次通过作业任务的各个位置，进行示教，同时将这些位置序列记录在存储器内。在任务的执行过程中，各个关节在伺服驱动下重新遍历出那些位置序列，从而完成作业。因此，这台原型机器人的主要技术功能就是“可编程”以及“示教—再现”。

20 世纪 70 年代，机器人产品正式问世，机器人技术开始形成。美国 Unimation 公司和 AMF 公司生产的工业机器人已用在汽车生产线，进行传送、上下料、焊接、喷漆等作业，表现出很好的灵活性和经济效益，推动了机器人的商品化。同时，机器人研究也不断地把机器人技术引向深入，如美国麻省理工学院研制的一种配有接触传感器的机器人装置，机器人凭触觉可以判断操作对象的形状，其人工智能实验室研究了如何实现具有人的手、眼、耳功能的机器人系统。美国斯坦福大学研制了带有电视摄像机和接触觉装置的移动小车，用于机器人运动规划问题的研究。上述这类研究工作逐渐形成了智能机器人的技术内容。

70 年代机器人产业蓬勃兴起，机器人学与技术发展为专门的学科。1970 年第一次国际工业机器人会议在美国举行。各种成功的实用范例推动了机器人应用领域的进一步发展。日本、德国、瑞典等国家的政府和产业界积极引进机器人技术，建立生产体系，大力推广应用，对机器人的发展产生较大影响。大规模集成电路技术的迅速发展以及微型计算机的普遍应用更使得机器人的控制技术出现了质的飞跃。

20 世纪末至今，随着计算机和人工智能技术的快速发展，机器人智能化成为可能。早期的机器人智能行为只能实现积木的分类、堆放动作等，一些有视觉和触觉的机器人可用于铸件、泵体的识别和检查，集成电路的装配和焊接。现在的机器人装有各种各样的内外传感器，对环境具有比较全面的感知能力，另外还有像筋肉一样的效应器，具有作用周围环境的能力。有的智能机器人已经能够很好地理解人类的语言，并可进行对话交流，而不仅仅是简单地模拟人或者动物的动作和行走，甚至可通过对环境的感知和识别，对情景的理解，进而对任务重新规划和实施。例如，美国波士顿 Dynamics 公司研制的“Bigdog”四足机器人（如图 1），它能够在冰面和险要斜坡快速跑动，并且在强大外力冲击时也能很好地保持本体的平衡。智能机器人的智力反映在它对环境信息的感知和处理以及对情势的判断和决策能力，或者说是一种对环境的、仿生的自适应能力。



图 1　四足机器人

一般认为，按照机器人从低级到高级的发展程度，可以把机器人分为三代。

第一代机器人，即工业机器人，主要指能以“示教—再现”方式工作的机器人。其本体是一类似于人的上肢功能的机械手臂，末端是手爪等操作机构。它通过操作人员“手把手”的示教，或者通过离线编程存储的动作顺序信息完成作业。如果外界条件发

生变化,或者机器人工作内容需要改变,必须通过人对其程序做相应改变,因此它不具备智能。

第二代机器人是指基于传感器信息来工作的机器人。它依靠简单的感觉装置获取作业环境和对象的简单信息,通过对这些信息进行分析和处理,作出一定的判断,对动作进行反馈控制。这类机器人初步具有智能。

第三代机器人,即智能机器人,这是一类具有高度适应性和自主能力的机器人。它本身能感知工作环境、操作对象及其状态,能接受、理解给予的指令,并结合自身认识外界的结果来独立地决定工作规划,利用操作机构和移动机构实现任务目标,还能适应环境的变化,调整自身行为。

由上可见,智能机器人是工业机器人从无智能发展到有智能,从初级智能水平发展到高智能水平的产物。

区别于第一代和第二代机器人,智能机器人必须具备四种机能:行动机能——施加于外部环境和对象的,相当于人的手、足的动作机能;感知机能——获取外部环境和对象的状态信息以便进行自我行为监视的机能;思维机能——求解问题的认知、推理、记忆、判断、决策学习等机能;人机交互机能——理解指示命令、输出内部状态、与人进行信息交换的机能。简言之,智能机器人的"智能"特征就在于它具有与外部世界——环境、对象和人相协调的工作机能。

围绕上述四种机能,智能机器人的主要研究内容包括:

(1) 操作与移动　随用途而异的各种机械臂结构及其驱动方式,包括具有冗余自由度的机械臂和主从机械臂;多指手及其他灵活的操作机构;轮式、履带式、单足跳跃或多足步行式、喷气或气垫式以及仿生式(如蠕动、爬壁)等各种适合环境的移动机构及其驱动方式。

(2) 传感器及其信息处理　基于摄像机的视觉传感器以及对二维、三维物体的信息处理;基于超声红外光、激光、雷达波、磁性装置等各种非接触传感器及其定位、识别、运动检测等信息的处理;各种接触觉、压觉、力觉、滑动传感器检测信息的处理;以及综合多种传感器信息以便作出合适估计和决策的传感器融合技术。

(3) 控制　对应于各种运动学、动力学建模、运动轨迹规划与生成以及基于运动学或动力学的控制方法;位置控制、力控制以及力与位置二者结合的柔顺控制方法;遥控作业等非结构环境下的各种主从控制方法;多机器人的协同控制问题;**专家系统**、**人工神经网络**、生物进化等新颖的智能控制方法;上述各种控制方法的仿真系统。

(4) 人机交互　用于离线示教和机器人自主行为的各种计算机控制与编程语言,包括动作、对象和作业目标水平级(任务级)三类;语音识别与生成;声、图、文的多媒体处理技术;视觉、力觉、触觉、声觉的临场感技术及其他虚拟现实技术。

(5) 体系结构　基于机器人整体工作原理,设计它的内部结构组成及其相应的管理控制方式。例如,分层递阶式体系结构——最上层组织级将给定的外部命令和任务分解为子任务或动作组合,传至下一层的协调级,产生一系列具体的动作序列,直至最下层的执行级从而形成行动机构的驱动指令,整个过程中各种传感器信号逐级向上反馈,经综合处理后实现决策。又如,基于行为的包容结构——若干基于传感的可以自由组合的独立单元直接建立从感知到控制动作间的映射关系,系统的行为由单元行为的时间、空间序列组成,并由仲裁机构监控。

(6) 机器智能　涉及具体的技术和方法,主要包括问题求解、自动规划生成、**模式识别**、**自然语言处理**、**机器学习**、**专家系统**、**知识库**等,另外还涉及人类智能的机理实质和人工智能的实现途径这两方面的基础理论研究。

(7) 应用研究　智能机器人的材料、能源等实用化研制问题;各行各业对智能机器人的应用需求分析和产业发展关系;智能机器人的使用所引起的人类社会心理和法律问题等。

智能机器人与人工智能学科密切相关。它一方面是人工智能技术与方法的典型应用对象,另一方面又是研究、开发人工智能技术与方法的试验床,其核心问题是研究如何实现从感知到行动之间的智能联系。计算机科学与技术的发展为智能机器人的工程实现提供了支持和推动力,智能机器人也可以看成是安装了各种拟人的、仿生的外部设备的计算机系统或装置。智能机器人与工业机器人可同时并存,但智能机器人实际上研究的是各种形态的智能机器,具有更为深入的学术意义和更为广泛的应用前景。

目前,智能机器人的研究还处于初级阶段,研究目标一般围绕感知、行动、决策思维三个问题。实验室原型有:自动装配机器人——具有对部件的三维视觉识别和定位,柔顺控制多指手爪的抓取和精密

装配，自动规划装配序列，避碰，多操作器协调等功能；移动式机器人——具有室内、外自主导航，路径规划，野外复杂环境下的移动，避碰，基于感觉的取样操作和检测、排除故障等功能；水下机器人——具有深水潜游，有缆遥控，水下清理和维修等功能。

上述各种机器人实际上只是理想的智能机器人的一部分功能环节，其研究的局限性主要在于人工智能技术至今还不能提供实现机器智能的有效原理和方法。智能机器人的完美实现需要提高机器的自主性，以进一步独立于人，建立更友善的人机界面；需要提高机器的适应性，以更好地适应环境变化，加强与环境的交互能力。

智能机器人的研究目前正在三个方面深入：依靠人工智能领域现有的成熟技术，发展面向专门任务的特种机器人；在研制各种新型传感器的同时发展基于多传感器集成的大量信息获取和实时处理技术；改变排除人的参与、机器人完全自主的观念，发展人机一体化的智能系统。

参考文献

1. 周远清，张再兴，许万雍，等. 智能机器人系统. 北京：清华大学出版社，1988

2. 张钹. 智能机器人的现状及发展. 科学导报，1992，6

3. 刘海波，李根深. 智能机器人研究现状及发展动向. 机器人情报，1991，5

4. Bekey G A. Autonomous robots：From biological inspiration to implementation and control. Cambridge，MA：MIT Press，2005 （张再兴　杨唐文）

zhinengka yueduqi

智能卡阅读器（smart card reader）　从智能卡上读出或写入数据的设备。智能卡也叫**IC 卡**，即**集成电路卡**，是在塑料基片上嵌入专用半导体芯片的卡。常见的 IC 卡大小为 85.6 mm×53.98 mm，厚度为 0.76 mm。移动通信系统使用尺寸较小的用户身份组件（SIM）卡，其大小为 25 mm×15 mm。

IC 卡由半导体芯片、塑料基片及通信接口组成，依照卡与外界数据交换的形式，IC 卡可分为接触式、非接触式和双界面三种类型。接触式 IC 卡通过卡上的电极触点与外界直接接触来交换数据，这种卡的通信接口为电极膜片。非接触式 IC 卡通过电磁波与外界进行数据交换，这种卡的通信接口为天线及相关电路。双界面 IC 卡同时拥有以上两种通信接口。

根据卡中半导体芯片功能的不同，IC 卡主要分为存储卡、逻辑加密卡和 CPU 卡。存储卡只有数据存储功能。逻辑加密卡是在存储卡的基础上增加安全控制逻辑，这在一定程度上可保护卡中数据的安全。CPU 卡内集成了中央处理器、存储器、加密协处理器、输入输出接口电路等，这种卡中一般固化有卡内操作系统（COS）。

IC 卡作为一种有效的身份识别工具和支付手段，与磁卡相比，具备以下主要特点：具有多层次的安全认证功能，不易伪造或复制，存储容量大，寿命长，抗电磁干扰能力强。

IC 卡阅读器是一个单片机系统，主要由 IC 卡接口电路、中央处理器（CPU）、CPU 外围扩展模块、加解密模块、电源模块及通信接口组成。IC 卡接口电路完成对 IC 卡的读写驱动。CPU 外围扩展模块可以是存储器、时钟电路、键盘电路及显示电路等。加解密模块可由专用加密协处理器构成。IC 卡阅读器与接触式 IC 卡及非接触式 IC 卡交换数据的接口分别是 IC 卡座和天线，与其他设备如计算机进行通信一般通过 RS-232C 或 USB 接口完成。

IC 卡阅读器从应用领域上可分为专用型与通用型。专用型 IC 卡阅读器通常是作为相关设备的一部分，并且在结构上是按一体化设计的，如 IC 卡电话机、IC 卡电度表、IC 卡门锁等。通用型 IC 卡阅读器可独立使用（独立型）或作为终端使用（联机型）。独立型 IC 卡阅读器拥有键盘和显示屏等人机界面，这种阅读器可自带微型打印机及网络接口，由于操作形式不同，它可分为台式和手持（便携）式。台式 IC 卡阅读器注重牢固、耐用，手持式 IC 卡阅读器强调轻巧、省电。联机型 IC 卡阅读器分为内置式和外置式，内置式 IC 卡阅读器通常安装在主机前面板的安装槽上，它可通过扩展总线与计算机进行通信；外置式 IC 卡阅读器一般放在工作台上使用，易于安装。

作为一种专用输入设备，IC 卡阅读器具有操作安全可靠、使用方便的特点。随着 IC 卡在电信、金融、税务、保险、交通、医疗、卫生保健、城市公用事业等多个领域的普及，IC 卡阅读器可得到更广泛的应用。

参考文献

王卓人，邓晋钧，刘宗祥. IC 卡的技术与应用. 北京：电子工业出版社，1999 （李炜）

zhinengti de BDI moxing

智能体的 BDI 模型（agent BDI model）　基

于智能体的思维属性建立的一种逻辑模型。模型使用信念(Belief,用 B 表示)、愿望(Desire,用 D 表示)和意图(Intention,用 I 表示)三个思维属性来描述智能体(参见**多智能体系统**),简称 BDI 模型。

对于智能体,通常从物理立场、设计立场和意识立场来描述。物理立场着眼于描述系统的物理特性和运动规律;设计立场着眼于描述系统的设计目标、模块功能等;意识立场则试图借鉴人类思维属性的概念,来解释和推理复杂系统的行为。意识立场把系统看作理性的智能体,通过信念、愿望、意图等思维属性对智能体的行为进行描述和推理。采用意识立场建立智能体思维状态模型对设计者和分析者来说是自然的,为描述复杂系统的行为提供了简洁的表示,便于理解和解释。而且,基于智能体的思维状态模型,不依赖物理实现就可以得到智能体行为的规则和模式,可被智能体用来进行互相推理。

智能体的 BDI 模型中,信念、愿望和意图等思维属性具有直观的含义。信念描述了智能体对当前世界状况、自身状况,以及其他智能体状况的认识,属于思维状态的认知领域。愿望描述了智能体对未来世界状况以及可能采取的行为路线的喜好,属于思维状态的情感领域,智能体可以拥有互不相容的愿望,甚至可以拥有当前求解能力下无法实现的愿望。智能体选择可以实现的愿望的子集作为目标,是智能体希望达到的状态。当智能体对某个目标作出承诺,准备付诸实施时,则形成意图。意图属于思维状态的意向领域,其作用是引导并监督智能体的动作。BDI 模型的哲学基础是 Bratman 对理性和意图的分析,刻画了意图的客观性以及意图在理性平衡中的中心位置。分布式人工智能领域的学者,基于 Bratman 的“意图中心论”,着重研究了信念、愿望、意图等思维属性的形式描述及其逻辑关系,从而形成了 BDI 模型。具体说,如果 Agent 是一个人,要做一些事,从意识观点看是说 Agent 想做些事(愿望),但有的是可行的有的是暂不去做的,即先选出真要做的事(目标),进而依 Agent 的能力、知识(信念)来采取行动(意图)实现目标,这就是 Agent 完成一件事的分解,采用 BDI 来表达。

智能体的 BDI 描述由澳大利亚学者 Rao 和 Georgeff 于 1991 年提出,其形式化系统是对计算树逻辑 CTL^* 的扩充,并使用经典的可能世界语义模型。

经典的智能体思维状态模型存在逻辑全知(logical omniscience)、无为而治(transference)和副作用(side-effect)等不合理的逻辑性质。针对这些问题,很多学者提出了改进的思维状态模型。同时,对其他思维属性(如义务、承诺等)也进行了研究。这些工作大都是基于 Rao 和 Georgeff 关于 BDI 模型的工作或在 BDI 模型提出之后进行的。如 Konolige 和 Pollack 基于非正规模态逻辑提出了意图模型,使用认知结构描述智能体的思维状态,引入了意图关系图的概念以直接的方式表示了意图的结构。Bell 提出了属性改变模型,建立了一套持续规则,表征智能体会保持它的思维属性直到有改变的理由,用模态连接词定义了影响、信念、愿望、意图、义务和理性等思维属性,并在模型中显式地加入了时间特性。Gaspar 和 Coelho 提出了目标和意图模型以及修正框架,扩展了智能体的信念模型。Linder 和 Hoek 讨论了喜好、目标和承诺等思维属性的性质,对逻辑全知问题进行了比较完善的处理。Cavedon 和 Padgham 指出了意图逻辑的非正规性,在 Kripke 框架中引入“不可能世界”,部分消除了正规模态逻辑给意图带来的不合理的逻辑性质。Georgeff 和 Rao 分析了意图修正的语义,通过在原来的 BDI_{CTL*} 逻辑中扩充了相信、愿望和意图的 Only 模态,给出了模型约束和公理,刻画了意图修正和信念修正、愿望修正的关系,为探索思维状态的动态演化指出了方向。Dongha 基于线性时态逻辑提出了承诺模型,给出了指导智能体对意图进行取舍的原则,也为智能体如何构造目标规划提供了依据。

参考文献

1. 石纯一,张伟. 基于 Agent 的计算. 北京:清华大学出版社,2007

2. Wooldridge M. 多 Agent 系统引论. 石纯一,张伟,等译. 北京:电子工业出版社, 2003

(张伟)

zhinengti tongxin yuyan

智能体通信语言(agent communication language) 一种用于表达智能体交互意图和内容,实现智能体间通信的描述性语言(参见**多智能体系统**)。它独立于智能体的具体实现技术和平台,具有严格的语法、语义和语用,支持参与交互的智能体对这些交互消息进行理解和分析,进而实现多智能体之间复杂的协作。智能体通信语言的语法是指智能体通信语言如何构成,语义是指交互消息中各种符号的指称,语用是指如何理解和解释由智能体通信语言所描述的消息,它们表达了什么样的协作意图,对参与交互的智能体产生什么样影响。

智能体通信语言通常需具备两方面的表达能力：①交互意图即语用，在多智能体系统中，参与协作的对象是一些具有认知状态以及拥有多样化知识、资源、信息等的自主行为实体，协作是智能体间复杂的交互过程。在此过程中，每一次交互都展示了参与交互智能体的不同协作意图，因此智能体通信语言应提供表达智能体协作意图的机制。例如智能体的交互是要请求对方提供服务，还是向对方提出一项建议；是要作出一项服务承诺，还是要向其他智能体提供某种信息。显然，这些协作意图的表述对于参与协作的智能体正确地理解对方的意图、有序地开展协作活动是重要的。②交互内容即语义，除了协作意图之外，智能体间的协作还涉及协作内容问题，也就是说要针对哪些方面开展协作。比如，一个智能体请求另一个智能体为它提供某种服务，那么服务的内容是什么。

智能体通信语言以言语理论（Speech Act Theory）为基础来表达交互的语用，借助于内容描述语言来表达交互的语义。言语行为理论主要研究和分析语言作为一种动作在人类或者智能体协作中的地位和作用。它强调人类交际的语言或其他表达手段，不仅仅是为了描述某种状态，而是为了完成一定的行为，比如“请求”，“通知”，“询问”，“允诺”等。言语行为的特点是：言者通过说一句话或若干句话来执行一个或若干个上述所列举的行为，这些言语行为的执行可能给听者或者说者带来某些后果。

为了支持多智能体系统的开发，目前人们已经提出了多种智能体通信语言，代表性成果包括：由美国国防部高级计划署（DRAPA）主持研发的知识查询与操纵语言 KQML（Knowledge Query and Manipulation Language），由智能物理智能体基金 FIPA（Foundation for Intelligent Physical Agents）提出的 FIPA ACL 语言。它们均提供了语法成分来表示交互的对象、通信行为等，并采用诸如 KIF（Knowledge Interchange Format）等来描述交互内容，此外 FIPA ACL 还提供了多样化的交互协议（Interaction Protocol）来支持智能体间基于交互的协同。

参考文献

1. Labrou Y, Finin T, Peng Y. The current landscape of agent communication languages. IEEE Intelligent System, 1999, 14(2): 45-52

2. Wooldridge M. An introduction to multiagent systems, John Wiley & Sons Ltd, 2001

（毛新军）

zhongduan

中断（interrupt） 计算机在执行程序过程中，当遇到急需处理的事件时，暂停当前正在运行的程序，转去执行有关服务程序，处理完后自动返回原程序，这个过程称为中断。

中断可分为内中断和外中断。内中断是由计算机内部原因引起的中断，如溢出中断、非法操作码中断、地址越界中断等；外中断指外部事件引起的中断，如输入输出中断、电源故障中断、实时钟中断等。外中断又叫强迫中断。在内中断中，由程序中特设的指令引起的中断，又称为软中断。

要求中断的请求可按其轻重缓急分级，并赋予一定的优先权，称为中断优先级。当有多个中断请求时，中断系统按中断优先级进行排队。排队原则是：级别高的优先响应。若在处理低级中断过程中又有高级中断申请中断，则高级中断可以打断低级中断处理，转去处理高级中断，等处理完高级中断后再返回处理原来的低级中断，称为中断嵌套。为了增加中断排队的灵活性，还可用程序的方法在某段时间中屏蔽某些中断请求，以改变中断响应顺序。有些中断请求是不能屏蔽的，如电源一旦掉电，中央处理器应立即响应，其优先级最高，称为非屏蔽中断。

中央处理器在响应中断后转入具体的中断服务程序之前必须保存其现场，包括程序断点、程序状态字和运算器中通用寄存器内容，以保证中断服务后能够恢复现场返回原来的程序。在保存现场和恢复现场的阶段，不允许任何新的或更高级的中断打断。系统采用“关中断”的办法禁止响应任何中断，等到保存现场或恢复现场完毕，再“开中断”。中断处理过程包括保存现场、恢复现场和具体的服务处理，都是通过程序实现的，因此这种方式又叫程序中断方式。

为了提高响应中断的速度，通常把所有中断服务程序的入口地址（或称中断向量）汇集为中断向量表。当中央处理器响应中断时，从中断向量表中直接得到相应的入口地址，并从该地址开始执行中断服务程序。

中断在现代计算机系统中是一种非常重要的技术，输入输出设备和主机交换数据、分时操作、实时系统、多处理机系统、计算机网络和分布式计算机系统中都要用到这种技术。

参考文献

谢树煜，刘风云. 计算机概论. 北京：机械工业

出版社,1987 （谢树煜）

zhongguo youlu wenti

中国邮路问题（Chinese postman problem） **图论**中一个有重要理论意义和广泛应用背景的问题。它来源于下述实际问题:“一个邮递员应如何选择一条路线,使他能从邮局出发,走遍他负责送信的所有街道,最后回到邮局,并且所走的路程为最短。”归结为数学问题,则是:“设给出了一个连通的无向图,它的每条边都有非负的长度,求 G 的一条经过每条边至少一次并且总长度最小的闭路径。”

上述问题是中国学者管梅谷于 1960 年最早提出并加以研究的,当时称为“最短投递路线问题”,并给出了解这个问题的第一个算法。

1965 年以后,国内外学者对中国邮路问题做了许多研究,并对原问题进行了许多推广与变形,以至于现在提到中国邮路问题时,指的是一大类问题,包括:

无向图上的中国邮路问题就是指上面提到的中国邮路问题。

有向图上的中国邮路问题指在一个有向图上,求一条经过每条有向边至少一次并且总长度最小的闭路径。这个问题相当于在所有街道都是单行线的情况下,求解最短投递路线问题。

混合图上的中国邮路问题指在一个既有无向边又有有向边的图上求一条经过每条无向边和有向边至少一次并且总长度最小的闭路径。这个问题相当于在部分街道是单行线的情况下,求解最短投递路线问题。

带风向的邮路问题指在一个无向图上,在下述假设下:“对于以 i 与 j 为顶点的边,从 i 到 j 和从 j 到 i 的长度可以不同”,求一条经过每条边至少一次并且总长度最小的闭路径。

乡村邮路问题指在一个无向图上,给定了一个边的子集,求一条经过给定子集中的每条边至少一次并且总长度最小的闭路径。

还有一些其他的推广,如有向乡村邮路问题,有容量限制的中国邮路问题等。对所有这些问题,均找到了不少算法,并对问题的复杂性进行了分析。已经证明,无向图与有向图上的中国邮路问题有多项式算法,而其他问题都是 NP 困难的。

中国邮路问题在近年来得到了广泛的应用,除用于邮政部门外,还曾用于扫雪车路线、洒水车路线、警车巡逻路线等的最优设计上。另外,从 20 世纪 80 年代起,人们又发现用计算机绘图时,如何节约画笔的空走问题,所对应的数学模型,恰好是中国邮路问题。而在计算机制造工业中,如将激光刻制用于集成电路加工的模具时,也可以用中国邮路问题来减少激光刻印机的工作时间。从而使中国邮路问题在计算机工业中也获得了重要的应用。

参考文献

Minieka E. Optimization algorithms for network and graphs. New York and Bacel:Marcel Dekker,Inc. , 1978 （管梅谷）

zhongwen xinxi chuli

中文信息处理（Chinese information processing） 利用计算机对中国语言文字及其所承载的信息进行加工、操作过程的理论、算法、技术以及应用系统的实现,或以之为研究对象的学科。中国语言文字泛指中国各民族所使用的语言文字。不过,由于汉语、**汉字**不仅在国内具有绝对优势,而且在国际上也得到广泛传播(境外亦称“汉语”为“华语”、“华文”),中国语言文字通常也特指汉语、汉字。本条目的中文即指汉语、汉字。语言文字是信息、知识和文化的主要载体。汉语、汉字本身也是中华文化的组成部分。以数字化手段实现中文信息在传统环境与由计算机和网络构成的虚拟空间之间的输入、输出、存储、传输以及检索、分析、生成和翻译等各种处理,是事关社会、经济、文化发展的具有战略地位的技术领域。与相关学科如语言文字学、计算机科学技术、认知科学、信息论、数学、哲学等的交叉融合,形成了中文信息处理这门新兴学科,并带动了以它为核心技术的产业的发展。

中文信息处理是影响信息技术使其具有民族文化特点的学科。20 世纪 50 年代有计划进行的中国语文现代化工作对后来中文信息处理技术的发展起到了奠基性的作用,如制定汉语拼音方案和推广普通话对于计算机的汉字输入和中文计算机的普及就至关重要。中文信息处理可划分为**汉字信息处理**和**汉语信息处理**两大部分,它们既有区别又有联系,相辅相成。

汉字信息处理是基础。在我国汉字信息处理具有一定规模的研究始于 20 世纪 70 年代中期。由于得到国家多项重点科技项目的支持,**汉字编码字符集**、汉字字型存储、输入、输出、编辑排版、**汉字识别**、速记等一系列核心技术取得重大突破,一批应用系统取得显著效益,对我国社会信息化进程和信息产

业的发展产生了巨大影响。在2000—2001年中国工程院和中国科学院联合评选的20世纪中国重大工程技术成就中，"汉字信息处理与印刷革命"被评为排名第二的重大成就。

与汉字信息处理通常以汉字和汉字串为处理对象不同，汉语信息处理则是指对词、短语、句子、篇章乃至语料库和网页等各类语言单位及其不同表达形式的处理技术。

汉语信息处理则以汉字和汉字串构成的词、短语、句子、篇章乃至语料库、网页等各级语言单位为处理对象。在汉字信息处理取得突破性进展的基础上，到了20世纪80年代中期，汉语信息处理研究取得了长足的进步。在我国，汉语信息处理研究实际上就是以汉语为核心的多语言信息处理技术。在20多年的发展历程中，既有理论方法的探索，也有诸如搜索引擎、**机器翻译**、文本校对、**信息提取**、言语识别、言语生成等应用系统的开发，特别是语言知识库等基础设施建设的成就令人瞩目。自2005年以来我国每年编纂发布《中国语言生活状况绿皮书》，这项工作需要处理10多亿字符的海量语料，如果没有中文信息处理技术的支持，这个任务是不可能完成的。

汉字信息处理一般不涉及汉字所承载的内容信息。汉语信息处理则不然，它要关注文本、话语所承载的内容，即传递的信息、表达的概念和知识，涉及语言的词法、句法、语义、语用，其最高境界是实现**自然语言理解**，难度大，至今尚未取得突破性进展。还要注意到，语言内容理解是更广泛的数字内容理解的难以分割的组成部分，必须依赖多模态信息的融合，必须仰仗脑科学、认知科学、信息科学、语言科学的共同进步和联合攻关。

随着我国综合国力不断增强，汉语在世界范围内已成为强势语言之一。一方面，中文信息处理技术已成为国际自然语言处理业界共同关注的焦点；另一方面，"国家中长期科学和技术发展规划纲要（2006—2020年）"将基于自然语言理解的中文信息处理列为前沿技术，昭示了中文信息处理技术在我国知识经济发展、国家安全维护、国际地位提升乃至人类科学探索中所占据的重要战略地位。

参考文献

1. 余锦凤，萧志春. 中文信息处理基础教程. 北京：北京大学出版社，2002

2. 宗成庆，曹右琦，俞士汶. 中文信息处理60年. 语言文字应用，2009，4：53-61

3. 俞士汶. 民族特点的文化要求——汉字汉语民族语言进入信息系统. 见：罗沛霖主编信息电子技术知识全书. 北京：北京理工大学出版社，2006

（俞士汶　黄昌宁）

zhongwen xinxi jiansuo

中文信息检索（Chinese information retrieval）　对包括文献、资料、网页在内的中文信息集合按一定方式进行组织、存储、管理，并根据用户的要求查找到所需信息的方法、技术和过程。它由计算机技术和图书情报学交叉形成。这里的信息集合不同于数据库系统当中存放的结构化信息，而是指非结构化信息，即现实世界中自然存在的没有经过规格化且带有歧义的信息，如一篇新闻、网页等。在20世纪90年代以前，"信息检索"被称为"情报检索"，后来由于信息来源不断扩充（从文献资料库到互联网），信息形式和信息载体不断丰富（从书目数据库到**超文本**、多媒体等），作为同义词的"情报"已让位给"信息"。

广义的信息检索既包括文本检索，也包括对语音、图像、视频等多媒体信息的检索，狭义的信息检索专指文本检索，中文信息检索则以中文文本为处理对象。

中文信息检索的历史发展

我国的中文信息检索研究始于20世纪70年代著名的"七四八"工程。80年代初开始提供国际联机检索服务。此后，我国自主开发的数据库（绝大部分是中文资料）和联机检索服务系统迅速增加。全文检索技术的发展缩小了中文信息检索和西文信息检索的差距。在Google搜索引擎以英文为主要语言在全球搜索市场取得领先地位的同时，百度搜索引擎成为全球最大的中文搜索引擎，百度目前采集的中文网页已超过50亿，接受来自世界各地的中文搜索请求，提供各种各样的搜索服务。

中、西文信息检索的区别

如果实现技术以字符串匹配为基础，中文信息检索和西文信息检索在原理和机制上并无本质区别。当今多文种处理已成为计算机系统普遍具有的能力，中文信息检索和西文信息检索实际上已经融合。但是，当信息检索技术与**自然语言处理**技术相结合向更高层次的智能化的概念检索发展时，中文信息处理的一些特点和难点就会显露出来。

除编码外，中文检索和西文检索的一个基本差

别是中文文本需要经过分词处理。由于中文文本中词语之间没有空格,无法通过空格间隙识别词语的边界,而**汉字**在表意方面并不完整,词语才是表达意义的基本单元,因此中文检索系统目前都要采取某种中文分词算法将中文句子切分为词汇序列。分词的准确率和速度直接影响中文检索的效果和效率。

如果要进行深层语义检索,则需要解决中文词义消歧、句法、语义分析等问题,由于和西文相比,中文缺少形式标记,因此目前中文的句法语义分析精度显著低于西文,中文深度语义搜索的困难较大。

此外,对于搜索引擎而言,中文搜索面对的主要是以中文为母语的用户,他们在文化上跟以西文为母语的用户有一定的差异,文化上的差异会导致搜索需求、搜索习惯以及对搜索结果满意度方面的不同。

信息检索的基本流程

信息检索的基本流程可以分为三个步骤:信息采集、信息加工和信息查询。

信息采集是把信息源的信息拷贝到本地的过程。信息源的种类非常多,可能是纸版图书,也可能是一台电脑里的DOC,PPT,PDF等各种格式的电子文档,还可能是某个企业内部的各类文件等等。而在信息检索中,我们最关注的信息源是互联网。因为互联网的信息是开放的,并且信息量巨大,而且还是不断更新的。在互联网上采集信息的软件被称为爬虫(crawler)。爬虫在网上搜索前进,每访问一个网页就把其中的内容传回本地服务器。网上存在大量垃圾页面,需要清理。此外,网页内还会有导航条、广告等与内容无关的信息,也需要通过网页分析去除,方便后续处理。

信息加工最主要的任务就是对采集到本地的信息编排索引,以便做好提供查询的准备。在传统的图书编目工作中,图书管理员需要对书籍进行分类、标引,并撰写摘要,这些被称为二次文献。信息加工的过程和图书编目的过程类似,但全部由计算机自动完成。由于网上的信息太多,鱼龙混杂,哪些信息具有可信性、权威性,需要一定的技术手段加以判别。Google最早使用PageRank算法,成功地对每个网页的价值给出一个评估值。这部分工作一般在信息加工阶段完成。

在信息采集与加工之后,就可以响应用户的查询了。用户输入查询式,可能是几个关键词的逻辑组合,也可能是自然语言的问句。信息查询系统接收该查询,转换为查询的机内表示形式,然后在索引表中快速搜索,找到与用户需求最匹配的若干文档,按照一定准则排序,将一部分结果返回给用户。用户对系统返回的检索结果进行浏览。如果用户给出哪个网页是他想要的、哪个不是的判断信息,这种信息就被称为反馈信息或者用户反馈。这种反馈信息提交给系统,检索系统就可以更加准确地理解用户的要求,从而给出一批更有可能满足用户当前需求的文档,这个过程叫做相关反馈(relevance feedback)。用户用什么形式表达需求,怎样理解用户的需求,怎样计算用户需求与文档之间的相关度,怎样呈现检索结果,这些都是该步骤的关键环节。

中文信息检索的研究内容

信息检索中的基础研究

(1) 信息检索理论与形式模型

对信息检索模型的研究是在抽象信息检索过程的基础上,采用四元组的形式模型(查询逻辑表示、文档逻辑表示、系统表示和排序函数)表示一个信息检索系统。采用形式模型研究信息检索的优点是,关注问题的核心和本质,忽略实现细节。模型的研究属于信息检索理论层面的工作,新的更为有效的模型的提出对信息检索的发展具有重大推动作用。

(2) 信息检索系统的体系结构

信息检索是应用性很强的学科,对它的研究不仅仅是在理论上,在实际实现上,也有大量问题值得关注。实现时,需要考虑系统性能、数据存储(例如是否压缩、存储格式等)、系统扩展性、系统体系结构以及检索的有效性。

(3) 内容表示

在信息检索分布化和网络化的趋势下,信息检索系统的开放性和集成性要求越来越高,需要能够检索和整合不同来源和结构的信息,包括支持各种格式化文件,支持多语种信息的检索,支持结构化数据、半结构化数据及非结构化数据的统一处理,和关系数据库检索的无缝集成以及其他开放检索接口的集成等。

(4) 信息检索评价方法与评测数据

"信息检索评价"指的是信息检索的性能评价,比如用准确率(precision rate)和召回率(recall rate)来衡量信息检索系统的性能。目前的搜索引擎系统更关心准确率。除了准确率和召回率之外,还有像R-Precision、MAP(mean average precision)、P@10等多种指标。有效的评测手段是推动信息检索进步的关键之一。

（5）文本挖掘

文本数据挖掘（text mining）是指从文本数据中抽取有价值的信息和知识的计算机处理技术。现在数据量越来越大，检索结果可能非常多。如果能利用数据挖掘技术快速、准确地从 Web 信息资源中抽取感兴趣的、潜在有用的模式及隐含信息，信息检索在此基础上进行索引和检索，则有可能大幅度提高检索效果和效率。

信息检索中的关键技术

（1）信息抽取

信息抽取研究如何从无结构或半结构化的数据中抽取有序数据，在信息检索领域有着重要的应用。无结构和半结构数据不容易被计算机处理，目前信息检索主要处理这类数据；如果数据是结构化的，则可以方便地导入数据库，为进一步处理提供便利。

（2）文本分类与聚类

在信息检索中，对信息的分类处理是提高检索性能的途径之一。**文本分类**的目的就是将文本自动分入已知类别中。文本聚类是指将文本集合分组成多个类或簇，使得同一簇中的文本内容具有较高的相似度，而不同簇中的文本内容差别较大。自动分类和聚类在信息组织、导航方面具有非常重要的作用。

（3）引用和链接分析

互联网包含了传统数据环境所没有的一种丰富信息，即互联网的超链接拓扑结构。网页间的超链接一方面引导网页浏览过程，另一方面也反映了网页创建者的一种判断，即有理由认为，如果网页 A 存在一条超链接指向网页 B，那么网页 A 的作者认为网页 B 包含了有价值的信息。目前链接分析可以被用来分析网页的重要程度，分析互联网中潜在的社区及之间的关联，理解互联网自身属性特点等方面。充分利用互联网的链接结构信息对互联网应用技术的研究将具有极为重要的意义。

（4）分布式信息检索

分布式计算能够利用地理位置上分布更广的多台计算机或者多个处理器的计算和存储资源解决大规模问题。利用分布式计算进行信息检索称为分布式检索。

（5）Web 信息检索

Web 具有海量数据，这些数据是动态增加的，数据格式多种多样，具有多种语言的信息；除了丰富的内容信息外，Web 的网页之间还有链接关系，即有复杂的结构信息。Web 数据的这些特点使其成为目前信息检索领域最具活力者之一。

发展趋势

中文信息检索未来的发展趋势主要包括移动化、个性化、智能化、社区化等，此外，多语言检索、多媒体检索也受到广泛关注。

（1）移动搜索是指以移动设备为终端，方便、快速、准确地获取信息资源。对互联网 WAP 网站上的信息进行收集和检索服务；使用文本摘要等技术把普通网页转换成适合手机小屏幕显示的展示方式；结合手机的定位功能提供基于位置的生活信息搜索服务。

（2）个性化信息搜索是以用户为中心的信息检索技术，它获取以多种形式表达的用户需求，分析用户检索内容和检索行为，全面、准确地描述用户的检索意图，并综合利用这些用户信息，提供能够满足用户个性化需求的检索结果。

（3）智能化搜索综合运用自然语言处理、机器学习、本体论等新的研究成果，使信息检索系统能够更准确深入地分析和扩展用户的查询意图，能够理解用户用自然语言句子发出的提问，并对查询结果文档进行文摘以及答案提取，使用户更便捷地获取结果，从而提高信息检索系统的智能化水平。

（4）社区化搜索通过对虚拟社区中的关系网络进行挖掘和发现，系统可以根据特定用户的背景信息和用户群的兴趣偏好，将相似用户获取到的信息相互推送，从而使相似用户之间协同式地进行知识获取与分发。

（5）多语言检索的基本目标是实现多种语言之间的交叉检索功能，比如使用中文检索词可以查询到英文相关网页信息。多媒体检索是指对图片、音频、视频等多媒体信息的检索。互联网存在着各种语言、各种媒体形式的信息，提高对多语言、多媒体信息的处理能力是未来信息检索的趋势之一。

参考文献

1. Frakes W B, Yates R B. Information retrieval: data structures and algorithms. Prentice Hall, 1992

2. 刘挺，秦兵，张宇，车万翔. 信息检索系统导论. 北京：机械工业出版社，2008

3. 李晓明，李星. 搜索引擎与 Web 挖掘进展. 北京：高等教育出版社，2003

（刘挺　邵艳秋　俞士汶）

zhongyang chuliqi

中央处理器（central processing unit, CPU）

在计算机内部对数据进行处理并对指令执行过程

进行控制的部件。中央处理器由运算器、控制器和处理器总线等组成。

指令执行过程

计算机在**程序**控制下进行信息处理,程序由指令组成,指令也像数据那样编码并存放在存储器中。程序运行时按控制器中的**程序计数器**所指地址,从存储器读出指令,经过指令译码器等部件的分析,向计算机各部件发出各种操作命令,完成相应指令的功能,最后得到程序运行的结果。

典型的算术逻辑指令的执行过程是:①形成指令地址,把指令地址送给存储器,发读存命令,读出本条指令。②取出指令送到控制器的**指令寄存器**,对指令操作码进行译码,决定执行何种运算或操作。③根据寻址方式形成操作数地址,把操作数地址送给存储器,发读存命令,读出操作数,送入运算器。④若需两个操作数可重复步骤③。⑤在运算器中对操作数进行操作码指定的运算。⑥形成运算结果地址,把结果地址和运算结果送给存储器,发写存命令,写入结果。若为寄存器-寄存器型指令,操作数在寄存器中,操作结果也放在寄存器中,除了取指令外不再访存,大大缩短了执行指令的时间。

控制器

控制器用来生成指令地址,取出指令,分析指令,向各部件发出一系列有序的操作控制命令,实现指令功能。在机器发生意外故障,或者有随机的输入输出请求时,采用中断方式终止现行程序的执行,转入中断处理程序,处理完随机请求后,又自动返回原来的程序。

同步控制与异步控制 计算机在执行指令的过程中,各种控制信号的时间关系是非常严格的,必须在准确的时间给出控制信号。控制信号如何产生,操作持续多长时间,下一个操作如何启动等问题,属于计算机的控制方式问题。

(1) 异步控制 当前的操作完成时,利用其完成信号作为后一操作的启动信号,这种由操作部件自己决定处理时间节奏的时序控制方式称为**异步控制**。

(2) 同步控制 控制器按固定的时间节拍给出控制信号来执行各种操作,这种控制方式称为**同步控制**。

如果各种操作时间相差不悬殊,而且又不随机变化,可采用同步控制方式。其特点是控制简单,易于实现。但为了照顾个别操作时间特别长(如乘法、除法运算),则可以局部地采用异步控制,但从全局看仍是同步控制。

控制器的基本组成 控制器包括:指令部件、地址部件、时序部件、中断控制部件和操作控制部件。

(1) 指令部件 指令部件包括:指令计数器(或程序计数器)——给出要执行的指令地址,保证程序自动连续地执行;指令寄存器——存放正在执行的指令,供控制器分析解释,直到该条指令执行完毕;指令译码器——把指令操作码翻译成某一个控制线上的有效电位,控制完成有关的操作。

(2) 地址部件 根据寻址方式形成转移地址或操作数有效地址。通常包括地址加法器,如在变址寻址时,把指令中的位移量与变址寄存器的内容相加,得到需要访问的主存地址。有时为节省器件或简化线路,也可将地址加法交运算器的**算术逻辑部件 ALU** 完成。

(3) 时序部件 给出有一定顺序时间关系的信号。一条指令执行过程可分为几个阶段,如取指令、执行等。每个阶段又可分成几步,每一步叫作一个节拍。节拍是时序信号的基本单位。节拍周期通常与时钟周期一致。启停电路用来实现启动、停机和单条、单拍操作,并控制这些操作时的时钟信号的输出。

(4) 中断控制部件 可以处理随机出现的各种意外请求,包括低速输入输出设备的输入输出请求。它不但提高了机器的工作效率,而且提高了机器的可靠性,并为分时操作、实时控制和网络通信提供了实现手段。

(5) 操作控制部件 是运行程序和执行指令的关键部件。为了指挥计算机各个部分协同工作,完成指令规定的功能,操作控制部件必须准确及时地向各部件提供各种操作控制信号。操作控制部件可以采用组合逻辑电路来实现,称为**硬连线控制器**。这种控制器的维护、调试、修改、扩充指令等都很困难,但因为速度比较快,为高性能计算机和精简指令集计算机所采用。除了硬连线控制器外,在很多计算机中采用微程序控制器(参见**微程序控制器**)。

运算器与数据通路

运算器是计算机的数据处理核心部件。主要由执行算术运算和逻辑运算的部件、存放操作数和中间结果的寄存器组以及连接各部件的数据通路组成。运算器按参与运算的各个二进位是由低到高依次运算还是高位低位同时运算,可分为串行运算器和并行运算器。串行运算器仅见于最早期的计算机。按参与运算的操作数是定点数还是浮点数,运

算器又可分为定点运算器和浮点运算器（参见**浮点运算器**）。

现代中央处理器中的定点运算器因为主要执行算术运算和逻辑运算，常被称为**算术逻辑部件**（缩写为ALU）。ALU是个处理中心，寄存器或存储器中的数据要送到ALU中进行处理，处理结果又送回到寄存器或存储器中去，有时各寄存器之间的数据传送也可以ALU为中心进行（这样可以节省许多连线）。这种以ALU为中心的数据处理和传送路径，就是中央处理器中的数据通路，如图1所示。

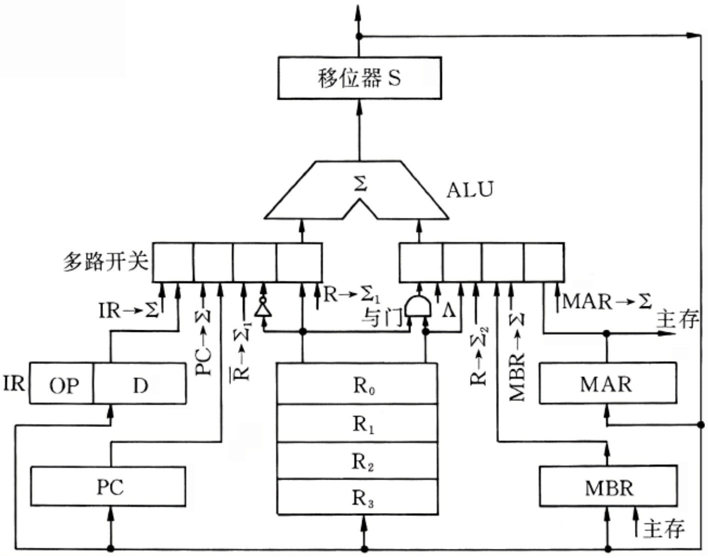


图1　以ALU为中心的CPU数据通路原理图

中央处理器的性能指标

执行指令的速度是衡量中央处理器性能的一个重要指标，一般用百万条指令每秒（MIPS）或十亿条指令每秒（GIPS）来度量。执行指令的速度一方面取决于中央处理器的时钟频率，另一方面取决于执行每条指令平均需要花几个时钟周期，而后者跟中央处理器的组织结构密切相关。在非流水线结构的中央处理器中，指令的执行过程是完全串行的，即一条指令经历取指、译码、执行的全过程后，再开始下一条指令的取指、译码、执行过程，这样，完成一条指令，就需要若干个时钟周期。流水线结构的中央处理器通过重叠相继指令的执行阶段，提高了指令执行的速度，在理想情况下，平均一个时钟周期就可以执行一条指令。超标量结构的中央处理器在流水线结构的基础上，通过同时开始多条指令的取指、译码、执行，进一步提高了性能，平均一个时钟周期可执行多条指令。

中央处理器的发展趋势

作为控制计算机各部分运行的核心装置，中央处理器经历了电子管、晶体管、中小规模集成电路、大规模集成电路等多次器件换代的变迁，从简单到复杂，从形式单一到形式多样。随着大规模集成电路技术的迅速发展，芯片集成度越来越高，中央处理器可以集成在一个半导体芯片上，称为**微处理器**，且功能不断增强。现代的中央处理器芯片除了包含定点运算部件、浮点运算部件、寄存器组、**向量运算部件**、地址部件、时序控制电路、操作控制部件、系统总线接口部件等传统上属于运算器和控制器的部件外，还包括**高速缓冲存储器**、存储管理部件、动态存储器控制器等传统上属于存储器的部件，甚至可能包括高速外部总线接口等传统上属于输入输出系统的部件。到2012年，单片集成多个中央处理器的多核处理器芯片已成为市场上的主流产品。单片集成大量中央处理器的众核处理器芯片也在不断地发展中（参见**多核处理器与众核处理器**）。

参考文献

1. 王爱英. 计算机组成与结构. 3版. 北京：清华大学出版社，2001

2. Patterson D A，Hennessy J L. 计算机组成与设计——硬件/软件接口. 原书第四版. 康继昌，等译. 北京：机械工业出版社，2012

（谢树煜　唐志敏）

zhongduan

终端（terminal）　参见**终端设备**。

zhongduan shebei

终端设备(terminal device) 通过通信线路或数据传输线路连接计算机的输入输出设备。简称终端。计算机终端设备通常设置在计算机中心以外更适合于该系统用户的地方。终端设备设置地点距计算机较远时,需在传输线路上加装调制解调器,这类终端称作远程终端。而终端与计算机距离较近,例如在同一建筑物内,不需经调制解调器传输的称作本地终端。

终端设备有许多类型,按用途可分为通用终端和专用终端。通用终端有交互终端与远程批处理终端两类。专用终端用于民航订票、旅馆房间预订、银行存取款、销售点收款记账、库存管理、教育、医疗以及其他方面,因业务的特殊性而有许多专门的类型,如销售点终端、银行窗口终端、自动取款终端、自学终端等。终端设备按功能完善的程度又可分为简易终端、灵巧终端和智能终端等。简易终端无缓冲能力,只能以低速异步传输方式工作,没有纠错能力。灵巧终端有数据缓冲能力,既可同步工作也可异步工作,有较高的传输速率,能检错和纠错,可对输入文本内容进行修改、编辑。智能终端备有微处理机、可编程序,既可联机运行,亦可独立工作,有较强的数据处理能力和文件管理能力。除上述终端设备类型外还有一些如小型便携式通用终端、光学字符阅读终端、汉字终端等。

交互终端 或称为分时终端,它可使计算机用户按分时操作与计算机对话,对每个单独请求作快速响应。每次向计算机发送一行请求、程序语句、数据。通常,交互终端的传输速率很低,因自终端发送的速率受到用户操作键盘速率的限制。但从计算机发至终端的信息则可以较快的速率传输。对智能终端、图形显示终端以及配备盒带驱动器及**软磁盘驱动器**的终端而言,则在两个方向均可以较快速率传输。通信连接往往是拨号电话式的。常见的终端设备类型有键盘打印终端与键盘显示终端两种。交互终端最早出现于20世纪60年代初,早期的终端设备主要是电传打字机。随着计算机交互应用的普及,许多专门设计的新型快速键盘印字设备陆续出现,性能显著提高,字符种类增多。显示终端明显优于打印终端。传输速率较快,一次可收发一行以上的电文,有些显示终端在发送到计算机之前可以修改所打的文本。显示部件因系电子部件,可靠性高。但显示终端不能留下永久的记录。为了弥补此缺点,通常在显示终端上配接印字设备。显示终端有字符显示终端及图形显示终端两种。

许多交互终端可接盒带驱动器、软磁盘驱动器等存储设备。在不与计算机联机的状态下,用户可将输入内容记录到磁带或软磁盘上,而后,再以较快速率将输入内容发送至计算机。同样,从计算机发来的输出亦可先记录在磁带或软磁盘上,然后再慢慢地由用户观察。

便携式通用终端 由**键盘**、小型印字机及音响耦合器等三部分构成。当电话线路与计算机接通后,将话筒耳机压在音响耦合器上,终端设备即可工作。音响耦合器是在终端设备与电话线路间靠音响耦合的一种调制解调器。键盘输入信号经音响耦合器变换成音响信号传送给电话机之受话器,然后经电话线路送至计算机。反之,计算机之输出同样以音频方式传至用户电话机之送话器,经音响耦合至音响耦合器,输出正常电脉冲信号,驱动打印机工作。

移动终端 通过无线网络,可以在移动中使用的计算机设备(参见**移动终端**)。

远程批处理终端 可使用户像在计算机中心一样,远距离地使用计算机。远程批处理终端所配置的输入输出设备与直接连接在计算机上的相同。最简单的批处理终端可以有一台行式印刷机和一台卡片输入机。复杂一些的终端还可以有卡片穿孔机、**磁带机**、磁盘驱动器以及其他输入输出设备。事实上,小型(甚至中型)通用计算机常用作大型系统的批处理终端。传输线路可以是专线,也可以是拨号电话线,典型速率为1 200~9 600 b/s,甚至更高。链接中需加调制解调器。简单的批处理终端不能同时收发,即只能半双向工作。而较复杂的批处理终端则可同时收发,即全双向工作。

专用终端 专用终端需专门的设计,以适合特殊业务的需要,便于操作人员尽可能方便和高效率地完成其各自的工作。这类终端通常用专线连接,而且是多路复用,即多台终端设备共享一台控制器及专线。这类终端高度专门化,可以不包含通用键盘或打印机,而代之以特殊的指示灯、信用卡阅读器、光学扫描器等。在有些情况下,它们还可提供以记录的或合成的语音形式的输出。

银行终端 有安装在银行办理存款、汇兑及贷款等业务窗口的出纳员用键盘打印终端及自动存取款机终端等类型。

出纳员使用的终端的基本构成包括:**微处理器**为主体的控制器、连接通信线路的接口、存储器、键盘、专用打印机、指示板等,此外还可选接字符显示

器、磁卡阅读器、磁条阅读器等。出纳员可依照显示器上显示的引导项目进行输入。存折可很方便地插入打印机的轨道中，自动到达打印的位置。终端并可读出存折磁条上的户主号码、存款余额、印字行数等信息。

由顾客操作的自动银行终端设备有自动取款机、自动存款机和自动存取款机（兼整钞换成零钞的功能）等三种类型，同时又有使用存折与不使用存折的区分。不使用存折的一种需使用磁卡。其基本结构包括：控制器、远程监视控制器、磁卡阅读器、存折插入、印字及退出机构、纸币分类保存箱与计数输出机构、纸币鉴别机构、纸币存入及不可分辨的纸币退还机构、结存便条打印及输出机构以及键盘指示器等。对三种类型的自动机器来说，不同之处在于取款机不需要纸币鉴别机构，而存款机又不需纸币计数输出机构。随着文字引导的显示还可伴随输出语音的解说。

教育训练终端 以 PLATO 计算机辅助教学系统中学生终端设备为例，在显示器屏上罩着一透明的特殊触摸键膜，学生利用这种触摸键可与计算机对话，跟随系统的引导进行各种作业的回答和进行各种实验。利用这种终端进行某些实验的训练还可免除危险，如化学反应及电力电网的实验等。

音频终端 由简单的语音输入设备和语音输出设备可构成音频终端。按键式电话机可作为一种音频输入终端，但键的种类仅限于 0 到 9 共 10 个数码。 （刘锡刚）

zhucong moxing

主从模型（master/slave model） 由一个主软件实体单向控制一个或多个从软件实体的分布式计算模型。如图 1 所示，主从模型的主软件实体负责接收计算任务；从软件实体接收主软件实体所给指令，完成所分配的工作。主从模型的典型示例是传统的**分布式数据库**，它通常有一个全局的控制模块，接受所有数据查询和更新的请求，再由该模块通知位于不同物理位置的子数据库完成查询和更新操作。

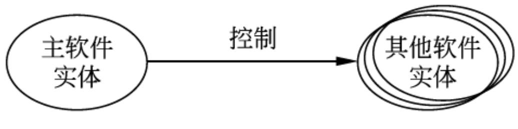


图 1 主从模型

主从模型的研究内容涉及架构建立、任务分解、任务协同等技术：①在架构建立方面，通过事先配置、动态协商等技术，主软件实体和其他软件实体建立起主从控制关系；②在任务分解方面，主软件实体接收计算任务，根据具体任务需求进行规划，分解后分发给模型中的软件实体；③在任务协同方面，主软件实体监视从软件实体的任务执行过程，接收从软件实体所返回的执行结果，并在必要时发出控制指令，避免死锁和其他冲突现象的发生。

主从模型具有结构紧密、协作高效的特点，已广泛应用于早期的**分布式软件系统**中。由于主软件实体位于系统的中心，其可靠性和性能是整个系统的瓶颈。再者，由于其开放性和可扩展性差，在广域、自治的分布计算环境下很难建立逻辑上紧耦合的控制与被控制关系，主从模型难以在互联网计算环境下得到更广泛的应用。 （吴泉源 丁博）

zhu cunchuqi

主存储器（main memory，MM） 中央处理器（CPU）可以直接访问的、存放当前正在使用的（即执行中）**程序**和数据的存储器，简称主存。目前，主存主要采用**半导体存储器**。主存的存取速度和容量对整个计算机系统的性能有重要的影响。

主存储器是按地址存取信息的。图 1 是主存储器的工作原理框图。它在工作时，中央处理器将地址送到地址寄存器（MAR），并发出“读出”或“写入”命令。地址译码器对地址进行译码，以选择相应的存储单元。主存接收到读出命令，则将存储单元的数据读出，送往存储器数据寄存器（MDR）；接收到写入命令，则将 MDR 中的数据写入存储单元。

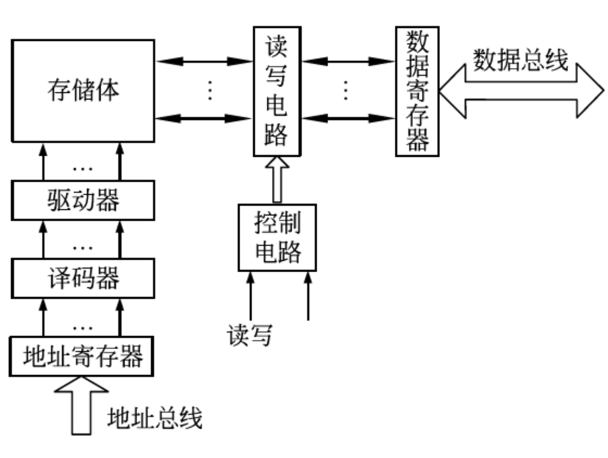


图 1 主存储器的工作原理框图

注：地址寄存器、数据寄存器逻辑上属主存储器，物理位置在中央处理器芯片中

通常，主存储器由**只读存储器**（ROM）和随机存取存储器（RAM）两部分组成。ROM中存储的内容只能读不能写，并且断电后信息仍保留，一般用于存放系统程序；RAM可读可写，但断电后信息会丢失，主要用于存储正在运行的用户程序和数据。RAM和ROM共享主存的地址空间。

由于单个存储芯片的容量有限，为了提供更大的主存容量，主存储器通常由一定数量的存储芯片组合而成。目前，主存储器常以内存条的形式实现。内存条是在一个条形的印制电路板上，用若干个存储芯片组成一定容量的存储模块。主板中通常有多个内存条的插槽，因此可以很方便地扩充主存的容量。按内存条的接口形式，常见内存条有两种：单列直插内存条（SIMM）和双列直插内存条（DIMM）。

为了提高主存储器的访问速度，出现了很多主存控制和访问技术。例如，同步动态随机存取存储器（SDRAM，synchronized DRAM）将CPU与RAM通过一个相同的时钟锁在一起，使CPU和RAM能够共享一个时钟周期，以相同的速度同步工作；双速率随机存取存储器（DDR，double data rage RAM）是SDRAM的更新换代产品，允许在时钟脉冲的上升沿和下降沿传输数据，这样在不提高时钟频率的情况下就能将速度加倍；存储总线式动态随机存取存储器（RDRAM，RAMBUS DRAM）在很高的频率范围内通过简单总线传输数据，并且能同时使用低电压信号，在高速同步时钟脉冲的两个边沿传输数据。

参考文献

1. 唐朔飞. 计算机组成原理. 2版. 北京：高等教育出版社，2008

2. 白中英. 计算机组成原理. 4版. 北京：科学出版社，2007

3. 王爱英. 计算机组成与结构. 4版. 北京：清华大学出版社，2007

（郑衍衡　陈妍　王换招）

zhuanjia xitong

专家系统（expert system）　一个智能程序，能对那些需要专家知识才能解决的应用难题，提供领域权威专家水平的解答。

专家系统主要由**知识库**、**推理机**和数据基三部分组成。知识库是知识的集合，包括待处理问题的领域知识，推理机实现问题求解过程中的推理。人类专家能够高效率求解复杂问题，除了因为他们拥有大量的专门知识外，还由于他们对知识的运用能力。知识的运用称为推理。具体而言，推理是指从已有事实推出新事实或结论的过程。推理机所采用推理方法与知识表示紧密相关。常用的推理方法有：基于规则的推理，基于逻辑的推理，**基于案例的推理**，贝叶斯网推理和证据理论中的证据组合等。

人工智能发展的前十年中，人们着重研究一般问题求解策略，如基于符号编程范式的定理证明程序和基于逻辑理论的通用问题求解程序。美国斯坦福大学E. A. 费根鲍姆（Edward A. Feigenbaum）发现一般性知识不足以使智能系统解决现实复杂问题，与领域专门知识的结合是必要的。基于此，1969年他和合作者研制了历史上第一个专家系统DENDRAL，能够利用质谱仪提供的信息对分子结构进行推断。DENDRAL的成功引发了专家系统在其他各个领域的成功应用，从20世纪70年代后期以来，相继出现大批应用于各领域的专家系统，主要代表有：用于医疗诊断的MYCIN系统，用于固体矿勘探的PROSPECTOR系统，以及用于计算机系统配置的XCON/R1系统。根据所求解问题的类型或特点，现有的专家系统通常可分为如下10类，如表1所示。与此同时，随着专家系统理论、方法不断地丰富和完善，派生出了**知识表示**、**不确定性处理**、**知识获取**、分布式人工智能等许多新的人工智能研究子域。

表1　专家系统的类型

专家系统类型	任务描述	特点
解释类专家系统	通过对已知数据信息的分析、解释，确定其内涵	要求系统能从不完备信息中得出解释，并对数据做出某些假设，系统需要对自身的推理过程给出解释
预报类专家系统	通过对过去和现在已知情况的分析来推断将来可能发生的情况	系统应提供随时间变化的动态模型，可以从不完全、不准确的信息中得出预报，并要达到一定的反应速度
诊断类专家系统	从观察的情况中推断出机能失常的原因所在	了解被诊断对象各组成部分的特点，互相联系，能够区分一种现象掩盖另一种现象的情况，能够提出需要检测的数据

续表

专家系统类型	任务描述	特点
设计类专家系统	在限定条件下给出目标的设计,任务是求出满足设计约束的目标配置	善于从多方面的约束中生成设计,易于对设计进行修改,能用以前的正确设计解释新设计
规划类专家系统	设计出能达到给定目标的一串动作	系统应能够抓住问题的重点,妥善处理各子目标间的作用和关系,处理不确定的数据信息,并通过一些试验性的动作得出可行的计划
监视类专家系统	将观察到的对象的行为与其应该具有的行为进行比较,以发现异常情况,给出警报	具有较快的反应速度,发出的警报有较高的准确性,系统随时间和条件的变化来动态地处理输入信息
控制类专家系统	自适应地管理一个对象的全部行为	能够解释当前的情况,预测未来,诊断问题的起因,不断修正计划和控制计划的执行
调试类专家系统	对失灵对象给出处理方法,推荐最佳解决方案	同时具有计划、设计、预报等专家系统的能力
教学类专家系统	根据学生特点和其知识及弱点,以最适当的方式进行教学和辅导	具有诊断、调试功能和良好的人机接口
修理类专家系统	对发生故障的对象进行处理,使其恢复正常	具有诊断、调试、计划、执行等能力

与传统问题求解程序相比,专家系统的主要特点包括:①具有显示表达的大量领域专门知识;②采用符号处理,将知识表示为符号体系,进而进行符号推理;③推理机、知识库分离和知识的模块性,使专家系统具有很大的灵活性;④具有自推理的能力,能够对系统自身的行为给出解释,向用户说明系统给出答案的由来。

专家系统的主要功能是模仿人类专家的智能活动,因此它需要在求解问题的过程中与用户进行对话。专家系统的人机交互包括两个方面:一方面专家系统要像人类专家在诊断疾病或诊断机器故障一样,向用户自动询问与问题求解相关的事实;另一方面,向用户解释它是如何求解问题的,或者推理出当前结论的依据。

当一个问题无法找到合适的求解方法时,专家系统有望采用领域专家的知识和经验加以解决。而人类专家的知识与经验往往呈现不确定性,待求解问题本身所提供的信息往往也是不精确和不完备的,因此专家系统应具有不确定性知识表示和处理的能力。

专家系统的结构通常有如下几种:①一般结构;②元知识系统;③分布式专家系统;④多知识表示结构;⑤多知识表示和多推理机结构;⑥多专家系统协同结构;⑦动态组织的体系结构。

图 1 给出了基于规则的专家系统的一般结构。如图所示,一个典型专家系统除了知识库、推理机和数据基三个主要部分之外,还包括解释器、人机界面和知识获取工具等部分。

数据基(data base)也称黑板(black board),用来记录系统推理过程中用到的控制信息、中间假设和中间结果。数据基主要包括如下三部分:①规划区,用于记录对当前问题总的处理规划、目标、问题背景和问题当前状态;②议程区,用于记录一些待执行动作;③中间解区,用于记录目前系统产生的中间结果和中间假设。

推理机(inference engine)是专家系统中实现基于知识推理的部件,是知识系统中不可缺少的重要组成部分,主要包括三个部分:①调度器,从议程区中选择下一次要执行的动作;②执行器,根据知识库中的规则,执行调度器选择的动作;③一致性协调器,对执行器新得到的假设做似然性修正。

知识库(knowledge base)主要存放和管理规则。规则通常以 IF A THEN B 的形式存在,其中 A 代表条件,B 代表结论或动作。

知识获取主要有以下几种途径:①非正式会谈,知识工程师与领域专家协作收集知识;②自动获取,主要利用基于**机器学习**和**数据挖掘**工具实现获取;③半自动获取。

解释器实现专家系统的**解释机制**,用于对推理的结论给出合理解释,主要采用执行追踪法、预制文本法、策略解释法和自动程序员解释法等。

人机界面是系统与用户进行交流的窗口。通过

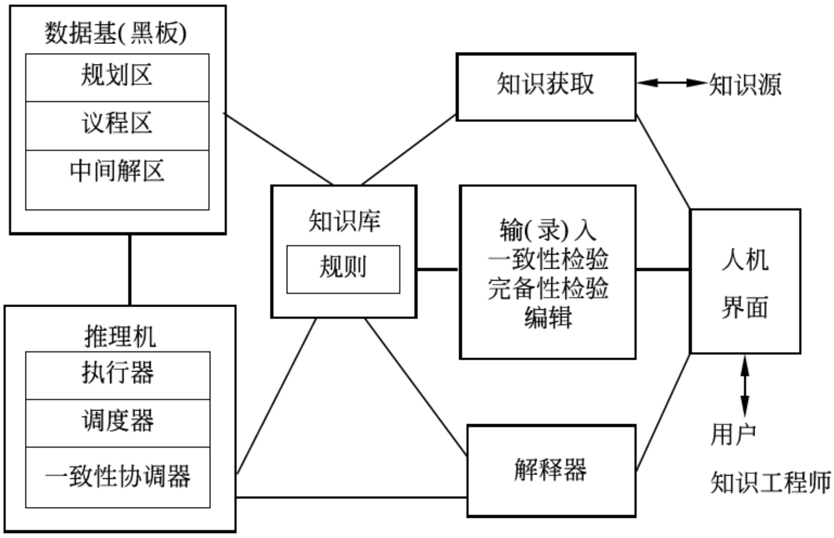


图 1　基于规则的专家系统结构

该界面,用户输入基本信息,对系统提出的相关问题进行回答,同时系统还会输出推理结果及相关的解释等。人机界面主要包括以下几种实现方式:①用户适应类(最低级):用户适应系统;②系统适应类(最高级):系统适应用户(个性化);③混合适应类(中间级):用户部分适应系统,系统部分适应用户。

设计和开发一个专家系统有很多原则可循,无论采用哪种原则,一般在一个开发周期内都需执行如下关键活动:问题分析,知识表示,知识获取,系统实现,系统测试与验证,评估和系统维护。

(1) 问题分析　专家系统解决的问题限于某个特定领域,待求解问题的恰当选择对于专家系统的设计和开发非常重要。待求解的问题大小需要适中,且需要考虑待求解问题是否需要符号推理,是否需要启发策略,是否需要依赖于不完整或者不确定的信息作出决策。

(2) 知识表示　将获取的人类专家知识表示成机器可理解的形式。主要的知识表示方法有:**产生式表示**、**语义网络表示**、**框架表示**、**逻辑表示**、**陈述性表示**、**过程性表示**、案例表示、**定性时空表示**和本体表示等。知识表示的关键问题是如何选择恰当的表示方法对特定问题进行表达。

(3) 知识获取　构造专家系统的关键任务是获取知识和建造知识库。可以从人类专家、文献等知识源获取待求解问题所需的领域知识。知识获取有很多途径,包括非正式会谈,采用全自动工具和辅助工具等。知识获取过程需要领域专家与知识工程师的协助。知识获取是构建专家系统的主要瓶颈。

(4) 系统实现　将获取的知识和已经表示完毕的知识转换成计算机程序代码。专家系统的实现方式通常包括:采用 Pascal 或 C++ 等高级程序开发语言实现;采用 PROLOG、LISP 等人工智能程序设计语言实现;采用专家系统外壳和**专家系统开发环境**等专门的开发工具实现。后一种方式可将开发者的精力集中在知识库的构建上,使得非计算机专业人员也能完成专家系统的开发。

一般刚建立的专家系统性能较差,可能由于对专家求解问题的行为理解不正确,或者由于抽取的概念有误,或者由于表达知识不正确,或者由于忽略了某些细节。这类错误并不完全表示专家和知识工程师专业水平不够,因为在设计系统之前,专家并不具有用有效方法表达知识的经验。事实上,无缺陷的开发过程是不存在的,需要通过实例检验,发现系统知识的脆弱性和缺陷,从而有针对性的改进知识库,逐步引导系统性能达到人类专家水平。这就要求专家系统开发除了以上介绍的四个主要步骤之外,还需经过系统测试、验证、评估及维护等阶段,以确保所开发系统的功能满足用户需求,运行结果的正确性,并根据用户需求的变化及时更新知识库中的知识。更新知识库中的某些知识通常会影响到其他相关的知识,需要保证知识库知识的一致性。

从第一个专家系统诞生至今,专家系统在理论上和应用上都取得了丰硕成果。专家系统的成功不仅标志着计算机处理对象从数值、数据处理阶段发展到了知识处理阶段,还标志着计算机的应用在深度与广度上向前迈进了一大步,跨入了计算机智能

化应用的新阶段。知识表示、推理、学习以及感知等和专家系统建造紧密相关，解决这些领域中的难题仍将是专家系统研究面临的主要任务。此外，随着信息技术的不断推陈出新和应用领域的不断拓展，专家系统建造所面临的问题和挑战也在不断变化。例如，如何从万维网这个巨大的知识源中自动获取领域知识并加以有效利用，成为万维网时代专家系统建造所面临的最大挑战之一。

参考文献

1. Russell S, Norvig P. Artificial intelligence: A modern approach. 3rd ed. Upper Saddle River, NJ: Prentice Hall, 2010

2. Giarratano J C, Riley G D. 专家系统原理与编程. 4 版. 印鉴，陈忆群，刘星成，译. 北京：机械工业出版社，2006

（刘大有　杨博）

zhuanjia xitong kaifa huanjing

专家系统开发环境（expert system development environment）　支持**专家系统**开发、运行和维护的软件系统，主要由专家系统开发工具和专家系统支撑环境组成。

专家系统开发工具

它是辅助专家系统开发的程序系统，一般可分为专家系统外壳、专家系统开发语言、专家系统部件开发与集成工具三类。

（1）**专家系统外壳**亦称为骨架系统，能提供**知识表示**、**知识获取**、推理控制和**解释机制**等功能，开发者只需加入领域知识，即可形成一个面向具体领域的专家系统，大大缩短了专家系统的开发周期。其体系结构如图 1。

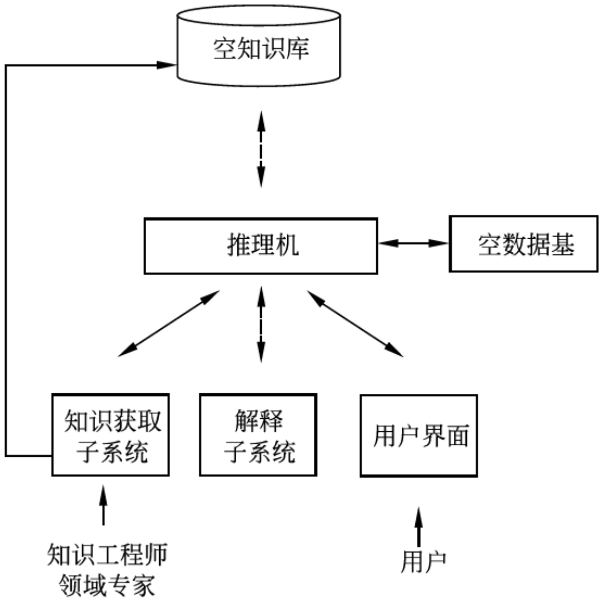


图 1　典型的外壳系统体系结构

早期著名的外壳系统有：从 MYCIN 系统分离出的 EMYCIN，适合解决诊断问题；由 PROSPECTOR 系统抽去地质勘探知识形成的 KAS，适于开发解释型专家系统；在青光眼诊断系统 CASNET 基础上开发的 EXPERT，用于诊断和分类；还有 AM，Crystal，Leonardo 和 EXSYS 等。外壳系统脱胎于具体专家系统，往往受限于原专家系统，只能解决特定问题，通用性、灵活性较差。

（2）**专家系统开发语言**是供知识工程师建造专家系统使用的支持知识处理的高级程序设计语言，也称知识处理语言。与常规的编程语言相比，该类语言在符号处理、模糊匹配、表示知识结构以及实施知识加工和基于知识的推理等方面均更具便捷性和有效性。使用该类语言可以方便地表示知识，利用知识进行推理、求解问题，并可以将它嵌入某种宿主语言中，从而使知识库、推理执行机构和解释机构的建立以及用户接口的实现都更为灵活（与外壳系统相比）、方便和有效（与常规的编程语言相比）。代表工作是卡内基-梅隆大学的 OPS5 等。

（3）**专家系统部件开发与集成工具**。开发者可针对具体领域问题，灵活选择工具，开发部件，并组装生成专家系统，比专家系统外壳更灵活，能开发多种类型的专家系统。主要有 Level 5 Object、ARTIM 和 LOOPS 等。

专家系统支撑环境

它是指辅助专家系统运行和维护的软件工具包，可为专家系统的开发提供多种方便的构件。代表性工作有：TEIRESIAS 和 TIMM 是两个典型的知识获取工具，能够帮助知识工程师把一个领域专家的知识植入知识库；其他工具还有辅助排错工具、知识库编辑工具、用户界面工具、解释程序等。

国内外主要的专家系统开发环境

国外早期的代表性专家系统开发环境有 KEE、CYC 等。1991 年美国 Intellicorp 公司开发了 PROKAPPA。1990 年，中科院数学所研制了“天马”专家系统开发环境。20 世纪 90 年代末至 21 世纪初，吉林大学、北京市农林科学院、中科院合肥智能所和哈尔滨工业大学在我国 863 项目的支持下，研制了一批农业专家系统集成开发平台。

参考文献

1. Joseph Giarratano, Gray Riley. Expert systems: Principles and programming 3rd ed. Boston: PWS Publishing Company, 1998

2. Keith Darlington. The essence of expert systems. Prentice Hall, 2000

3. Simon Kendal, Malcolm Creen. An introduction to knowledge engineering. Springer, 2007

（刘大有　王生生）

zhuanyong jicheng dianlu

专用集成电路（application specific integrated circuit, ASIC） 按照用户的特定要求而专门设计制造的一类集成电路，通常指专用数字逻辑集成电路。使用专用集成电路可使电子系统具有独特的功能和性能，且不易被仿制。同时由于设计方法的进步与设计工具的完善，其开发周期已大为缩短，开发成本也大幅度降低。

按设计方法不同专用集成电路可分为三种：①定制电路，如标准单元电路、全定制电路；②半定制电路，如门阵列（GA）；③可编程逻辑电路，如可编程逻辑器件（PLD）、**现场可编程门阵列**（FPGA）。

专用集成电路的开发方法取决于用户对性能、费用和开发周期等需求的综合考虑。定制电路一般开发周期略长，但性能最好，当生产批量大时性能价格比最优。门阵列开发周期较短，适用于中等用量的用户。可编程逻辑电路开发周期最短，适用于需要量较少的用户。

专用集成电路广泛用于计算机及各种外部设备，同时也广泛应用于通信、工业控制、武器装备以及消费类电子产品（如洗衣机、电冰箱、电视机、照相机）中。

可编程逻辑器件（PLD） 可由用户通过编程实现特定逻辑功能，具有一定通用性的专用集成电路。用户可按照自己的设计要求，利用专门的开发工具，对 PLD 进行编程以构成专用的逻辑集成电路。

PLD 由编程元件阵列与编程电路、寄存器、输入输出电路等组成。它的逻辑功能主要由产生乘积项的可编程“与”门阵列与产生和项的固定的“或”门阵列来实现。PLD 内部的编程元件可以是熔丝，或紫外光可擦编程只读存储器，或电可擦编程只读存储器。采用熔丝作为编程元件的 PLD 只能一次性编程，不能改写。采用紫外线可擦可编程只读存储器（UV EPROM）作为编程元件的 PLD 其封装带有透明窗口，以便用紫外光擦除，再用电气方法改写，从而允许多次编程。采用紫外线电可擦可编程只读存储器（UV EEPROM）作为编程元件的 PLD 可用电气擦除和改写，允许大量重复编程。

PLD 的开发周期短，广泛用于电子电路的验证，适于小批量生产。

通用逻辑阵列电路（GAL）是可编程逻辑器件的一种，其内部结构基本上与 PLD 的相同。它采用 EEPROM 作为编程元件，允许多次重复编程。

门阵列 利用预先制作的由大量器件规则排列成阵列的基片，按照用户要求进行连线封装等加工制成的专用集成电路。门阵列电路包含基片、宏单元和布局布线三个组成部分。基片是用基本单元按照一定规则排列而成的阵列，基本单元则是未经连接的一组晶体管。每个宏单元对应于一组特定的连接线，它可以将一个或几个基本单元中的晶体管连接起来构成具有某一特定功能的逻辑电路，如门电路、触发器、加法器等。布局布线的作用就是按照用户逻辑图的要求，确定宏单元的种类、数量及它们在基片上占有的位置，并确定这些宏单元之间及它们与芯片的输入输出引脚之间的连接。

门阵列基片独立于用户的特定需求，它可由专用集成电路制造厂家批量预先制作。应用时，根据用户的逻辑图，仅需在工作站上模拟验证，通过布局布线形成包含宏单元及其相互连接的文件，并以此为基础制作出互连掩膜，按照这些掩膜在基片上形成连接线，经过封装、测试，即可完成专用集成电路的制作。

门阵列电路具有开发周期较短（一般为 4 至 6 周）、开发成本较低的显著优点，在计算机、通信等电子系统中普遍使用。

现场可编程门阵列 一种可由用户通过现场编程实现特定功能、面向多用户多应用的逻辑门阵列电路。用户可根据其电路设计要求，选购相应集成规模、性能的 FPGA 产品，利用专门的开发系统，经过模拟验证，布局布线，生成编程文件，对选购的 FPGA 进行编程，以开发成用户需要的专用集成电路。

标准单元逻辑电路 采用标准单元方法设计制造的一类专用集成电路。标准单元是与数字逻辑电路中的各种门电路、触发器、寄存器、加法器等相对应的集成电路设计单元。一个标准单元具有三种形态，即用于逻辑模拟的逻辑符号（含功能描述），用于性能验证的延时参数以及用于版图布局布线的单

元版图。延时参数与专用集成电路制造厂家的工艺参数、电学参数密切相关。数以百计的这种不同的标准单元就构成了一个完整的标准单元库。

根据用户的高层电路描述或逻辑图,设计工作站上运行的逻辑综合软件可从标准单元库中调出适用的标准单元,把原始设计转换成标准单元之间的连线图(称为网表),并进行逻辑模拟和验证。设计验证通过后,即可用标准单元进行布局布线,形成用来控制电路制造的完整版图。再通过工艺加工、封装、测试,完成标准单元逻辑电路的制造工作。

由于标准单元通常都经过优化设计,在布局布线版图中无冗余单元,因而性能较好,占用的芯片面积也较小。虽然开发成本略高,但当生产批量较大时,采用标准单元法开发专用集成电路仍是最佳选择。

标准单元逻辑电路在计算机、通信等电子系统中应用广泛,有很大发展前景。

参考文献

Michael John Sebastian Smith. 专用集成电路. 虞惠华,等译. 北京: 电子工业出版社,2007

(仇玉林)

zhuanyong jisuanji

专用计算机(special purpose computers)

为解决一个或一类特定应用问题而设计的**计算机**。专用计算机的硬件和软件的配置依据解决特定应用的计算问题的需要而定,并不求全。现代专用计算机硬件大都采用冯·诺依曼体系结构,配有解决特定应用问题的专用程序。和**通用计算机**相比,专用计算机具有简单性、经济性以及解决特定应用问题的高效性等明显特点。

专用计算机大致可分为两类:一类是为完成一个特定应用任务而设计的专用计算机,如用于汽车点火装置控制系统的专用计算机、用于机床操作过程控制系统的专用计算机等,这类专用计算机嵌入到设备中使用,应用功能单一。嵌入设备中的专用计算机一般称为微控制器或**嵌入式计算机**。另一类是为完成一类特定复杂应用计算处理任务而设计的专用计算机,如用于高效处理多种图像处理算法的专用图像处理机、用于高效运行特定对象的复杂过程模拟计算程序的专用仿真计算机等,这类专用计算机尽管具有一些通用计算机的特征,而且是图灵完全的,但它们是根据一类特定应用问题的需求设计的,应属于专用计算机范畴。

专用计算机的历史比通用计算机要早得多。早期的**模拟计算机**是主要用于连续系统仿真的专用计算机。20 世纪 40 年代通用电子数字计算机出现之后,由于专用计算机在解决特定应用计算问题上的简单性、经济性和有效性,仍然不断发展。1943 年英国研制的用于解译德军密码的 Colossus 计算机,被称为是世界上第一台专用电子管数字计算机。文字处理系统是历史上曾经非常流行的一种专用计算机,许多公司都研制销售过这种严格只用于文字处理的电子管专用数字计算机和专用晶体管数字计算机。早期的专用计算机在创造了经济效益的同时,也促进了计算机的发展。

20 世纪 70 年代,由于集成电路的出现,通用计算机的性能显著提高,使得许多特定的复杂应用问题的高效计算处理,利用现代通用计算机也可以圆满完成。但对那些要求很高的实时计算性能和要求很高的长期运行可靠性的特定复杂应用问题,利用通用计算机在成本和效率上往往难以满足需求。当时,面向一类特定复杂应用问题的专用计算机,仍然得到不断的发展,如美国采用集成电路研制的用于导弹系统实时仿真的仿真机 AD10、AD100 等。20 世纪 80 年代后,精简指令集(RISC)微处理器的问世与高性能并行处理计算机体系结构技术的发展,使面向特定应用的高性能专用计算机的研制,普遍采用在通用并行计算机中增加专用的硬件加速部件和开发相应专用软件的技术路线。如美国的弹道导弹防御系统专用并行仿真机 CrayXD1、海下战争实验专用并行仿真机 SP tempest、分子动力学专用并行仿真机 Anton 等。

20 世纪 80 年代,嵌入式微处理器技术的出现和迅速发展,使各种嵌入设备中的专用计算机所要解决的特定问题,采用在嵌入微处理器上运行专用程序来解决,变得更为简单、经济和高效。现在,许多信息家电和工业设备,如移动电话、录像机、自动点火装置、机床等,都包含有采用嵌入式微处理器的嵌入式计算机。这些嵌入式计算机有的是可编程的,是图灵完全的(如个人数字助理 PDA 等);但更多情况下,当设计完成后,大部分程序被存储在只读内存中,或固化在硬件中,这样的嵌入式计算机已不是图灵完全的(参见**通用计算机**),但具有简单、高效、实用、可靠的特点。

随着系统集成芯片 SOC(system on a chip)技术的迅速发展,已可将一个具有复杂功能的嵌入式计算机系统集成到一个芯片上,使嵌入式计算机系统

的速度更快、尺寸更小、功耗更低。现在,有许多信息电子设备中已采用 SOC。嵌入式计算机系统向 SOC 转变,不仅是一种概念上的突破,而且是计算机和微电子技术融合发展的必然结果,将引起人类社会生活的新变革。

参考文献

1. 赵明主编,史国川主审. 计算机应用基础. 北京: 清华大学出版社,2009

2. Harver G cragon. Computer achitecter and impiemetation. Cabridge University Press, 2000

3. 胡守仁. 计算机发展史(一): 早期的计算器与电子管计算机. 长沙: 国防科技大学出版社,2004

(李思昆)

zhuanhuan jiance huanchongqi

转换检测缓冲器(translation lookaside buffer,TLB) 在**虚拟存储器**系统中实现虚实地址快速转换的硬件机制。它实际上是专门用于页表或段表(参见**虚拟存储器**)的**高速缓冲存储器**,所以也称为**快表**,现以页式虚拟存储器为例加以说明。在指令执行过程中,取指令时需要将虚地址转换为实地址,然后再根据实地址到内存中取指令,如果这条指令中含有访存操作,那么还需要一次或多次虚实地址变换操作后,才能到内存中访问数据。因为每次地址变换都需要访问页表,而页表较大,只能放在主存中,所以每执行一条指令,就会因为虚实地址变化的需要,而增加一次或多次内存访问。因此,如果没有由硬件支持的快速变换装置,虚存系统会因为低效率而难以实用。

正如高速缓冲存储器和虚拟存储器的工作都是基于程序的访存局部性,页表的访问也存在着局部性,而且局部性更为明显。因此,一种简单的方式就是让页表项进入数据高速缓冲存储器,以节省地址转换的周期数;但即使这样,有效地址也需要 1 个周期后才能得到。另一种简单的方式是设置一个专门的寄存器,用它记住上次转换的虚页号和实页号,从而当下次访问针对同一页时,不必再进行转换。

这种转换寄存器的更一般形式是由硬件提供专门用于存放页表项的高速缓冲存储器,即 TLB,有时也称为地址转换高速缓冲存储器(ATC)或目录检测表(DLAT)。TLB 的工作机制同高速缓存完全相同。因为 TLB 相对较小(通常含 64 ~ 1024 项),所以其访问时间极短,从而只要 TLB 命中,实地址就可以在 1 个周期内得到。如果 TLB 缺失,则地址转换仍需经过内存中的页表(也许访问页表还会导致缺页),且 TLB 的内容也必须更新。由于访问页表项的局部性很强,加之 TLB 总是采用有助于提高命中率的设计方案(如组相联或全相联映射、最近最少使用替换策略等),TLB 的命中率极高,一般都超过 99%。另外,当 TLB 缺失时,经常能在数据高速缓冲存储器中找到所要的页表项。

鉴于 TLB 对虚拟存储器系统性能的影响,20 世纪 80 年代以来所有的虚拟存储器系统都采用了 TLB(早期系统经常把 TLB 放在管理虚存系统的存储器管理部件 MMU 内)。为了进一步减少进程上下文切换时清除 TLB 内容带来的性能损失,一些计算机系统还为系统态和用户态各设置了 1 个 TLB,另一些系统则在 TLB 中加入了进程号的相关信息。

在采用分离式指令和数据高速缓存的系统中,经常为指令访问和数据访问设立专门的 TLB。近年来的系统为了应对由于存储容量和页表项数目日益增长而导致的 TLB 命中率降低问题,在设置高速小容量的一级 TLB 的基础上,开始引入容量较大但访问速度稍慢的二级 TLB。

参考文献

1. 郑纬民,汤志忠. 计算机系统结构. 2 版. 北京: 清华大学出版社,2004

2. Hennessy J L, Patterson D A. Computer architecture: a quantitative approach. 5th ed. Morgan Kaufmann, 2011

(唐志敏)

zhuangru chengxu

装入程序(loader) 把保存在外存介质上的代码(通常是目标程序)装入内存并启动执行的程序。

根据目标程序的形式,装入程序可分为三类:绝对装入程序、再定位装入程序和连接装入程序。如果待装入的目标程序中的所有地址都是绝对地址,则装入过程是根据装入地址、启动地址及目标程序大小等信息,将目标程序读写到指定的内存位置,从指定的启动地址开始执行,完成这种过程的程序称为**绝对装入程序**。如果待装入的目标程序中的地址是相对地址,则可把目标程序置于内存的任何位置,装入程序根据存取方式的不同调整目标程序中的地址,这种装入程序称为**浮动装入程序**。**连接装入程序**结合了浮动装入程序和**连接编辑程序**的功能,它把许多分别编译后的目标程序组合成一个可执行程序,再装入内存并启动执行。

装入程序可以是一个独立的程序，也可以是操作系统的一个部分。装入程序本身需驻留在内存，它的装入是由引导程序完成的。

一般装入工作是在整个程序执行前完成，但是有些情况下需要在程序运行中再装入某些程序。可在程序执行中装入新的目标程序并将控制传递给该程序的装入程序称为**动态装入程序**。

参考文献

Beck L L. System software: an introduction to systems programming. 2nd ed. Addison-Wesley, 1990

（张素琴）

zhuangtai zhuanyi tu

状态转移图（state transition diagram） 一种表示系统或系统元素的状态及其转移的图形工具。又称状态图。其基本构造包括状态、事件和状态转换。状态是由一组属性定义的，表示系统或系统元素所处的阶段、可实施的活动以及可见的特征；事件是由激发状态发生变迁的条件与活动定义的，通常是由消息规约的；状态转换是状态之间的一种关系，例如顺序关系、并发关系等。

状态转换图是一个有向图，图中的结点表示系统或系统元素的状态，有向边表示状态转换，可以在每条边上给出引发状态转换的事件。

在实践中，可以采用状态转移图来表达系统的行为模型。为了控制表达系统或系统元素动态行为的复杂性，可以采用分层的状态转移图。

（李宣东　王立福）

zhunru kongzhi

准入控制（admission control） 面向连接的网络中实现服务质量（QoS）保障的一种基本机制。主要功能是判决网络是否有能力接纳具有一定服务质量要求的连接请求。在分组交换网络中，提高资源统计复用率的技术目标与满足用户服务质量要求通常是相互冲突的，准入控制有利于在两者之间取得期望的平衡与折中。

一般而言，准入控制仅工作在面向连接（或虚连接）的网络中。用户产生新的连接请求，在协商阶段，网络根据当前可用的资源和连接请求中用户业务流量特性的描述，在测量或模型分析等技术手段的支持下，对是否接纳该连接请求做出决策，决策依据来自对下面两个主要问题的评估：

（1）网络是否有充分的可用资源为该连接提供满足其要求的服务质量？

（2）接纳该连接请求是否会影响网络保障已有连接的服务质量？

评估过程中对上述两个问题的任何否定性结论都将使网络拒绝该连接请求，并通告业务请求的发起者。如果评估过程接纳了该连接请求，网络需要为该连接分配能满足其服务质量要求的资源。

准入控制的评估方法大致有测量和模型分析两种，前者试探性地向网络中注入连接请求约定的业务流量，测量相关性能参数来完成决策；后者建立业务统计模型，通过求解复用队列模型来量化分析接纳新连接后系统的性能参数，参照相应的分析结果做出判决。

参考文献

1. Schwartz M. Broadband integrated network. Englewood Cliffs, NJ: Prentice-Hall, 1996

2. Perros HG, Elsayed KM. Call admission control schemes: A review. IEEE Communications Magazine, 1996, 34(11): 82-91

（任丰原）

zhuntongbu shuzi tixi

准同步数字体系（plesynchronous digital hierarchy, PDH） 一种在**同步数字体系**（SDH）出现之前被采用的数字传输体系，主要是为语音业务而设计，适合点对点传输，缺乏全球统一的标准数字信号速率、帧结构和线路接口。

数字传输系统通常都以时分复用方式进行传送，如果被复接的支路信号的时钟来自同一个时钟源，而且被复接的各支路信号与本机定时信号是同步的，这样的支路复接称为同步复接；如果被复接的支路信号的时钟来自不同的时钟源，各支路信号与本机定时信号是异步的，这样的复接称为异步复接。为了确保通信质量，在异步复接中，如果要求不同时钟源的差别不超过规定的范围，即各支路信号的码率只能在标称值附近有偏差，则称为准同步复接，PDH 就是基于准同步复接的传输技术。

PDH 有两种基础速率，一种以 24 个 64Kbps 信道构成的 1.544Mbps 为第一级（一次群，或基群），在北美称为 T1；略有变形后在日本采用，成为 J1。另一种以 32 个 64Kbps 信道构成的 2.048Mbps 为第一级（一次群，也称为基群或 E1），被欧洲和中国采用。PDH 除了基群信号采用同步复接外，由 2 的倍数个基群构成的其他高等级（也称为高次群）数字

信号信道采用异步复接。PDH 的速率等级主要有三种体系,其中欧洲体系为:2 Mbps,8 Mbps,34 Mbps,140 Mbps,565 Mbps,…;北美体系为:1.5 Mbps,6.3 Mbps,45 Mbps,274 Mbps,565 Mbps,…;日本体系为:1.5 Mbps,6.3 Mbps,32 Mbps,100 Mbps,400 Mbps,…。

PDH 没有全球统一的标准,各厂家自行开发的线路码型和接口各不相同,难以实现互连互通,而且组网和管理能力弱,无法适应传输网大规模发展的要求,因此 PDH 逐步被技术更先进的 SDH 所取代。

参考文献

林达权. 数字光纤通信设备(PDH 部分). 西安:西安电子科技大学出版社,1999

(*唐雄燕　马严*)

zichengxu

子程序(subprogram)　与子计算任务相应的处理对象和处理规则的描述。它是一个可被其他程序(单位)调用的程序单位。例如过程,函数,子例程。

在程序设计语言发展的早期,已认识到有些程序要在不同位置多次使用。如对数函数,各类三角函数等。为方便起见,就要求此类函数一经程序人员编码,就可在任何所需的地方使用。

子程序包括定义和调用两个方面。前者是定义子程序算法,后者是子程序的使用。下面用一个 ALGOL 语言程序的例子来说明过程(即子程序)的定义与调用。

```
begin
  array B[1:100,1:100],C[1:50,1:50];
  procedure TRANS(A,N);
  array A;integer N;value N;
  begin integer I,J;real TEMP;
     for I:=1 step 1 until N do
       for J:=1 step 1 until N do
         begin
            TEMP:=A[I,J];
            A[I,J]:=A[J,I];
            A[J,I]:=TEMP
         end
  end;
  ……
过程调用              TRANS(B,100);
    ……
过程调用              TRANS(C,50);
    ……
  end
```

过程 TRANS 定义为求一个 N×N 矩阵的转置矩阵。在 ALGOL 等语言中,矩阵称为数组。执行过程语句 TRANS(B,100)将 B 转置,结果仍在 B 中。在 ALGOL 语言中,若在 Procedure 前冠以类型名,则此过程为函数过程,此函数过程体中应有对此过程名的赋值。函数过程调用(ALGOL 60 中称为"函数命名符")可出现在表达式中,这与通常所说的过程不同。

在 FORTRAN 语言中不称过程,而称子程序。FORTRAN 子程序可分为两类,即 SUBROUTINE 子程序和 FUNCTION 子程序。函数(子程序)的结果值在函数体中必须被赋给它的名。而子例程(子程序)则否。它只能通过 CALL 语句,如 CALL TRANS(B,100)来调用。不能在赋值语句的表达式中调用。

在 Ada 语言中,子程序定义为过程或函数。除在书写格式上稍有差异外,Ada 过程本质上类似于 ALGOL 过程,函数则类似于 FORTRAN 的函数子程序。Ada 还允许参数默认,即在调用时可不指定相应的实在参数,而实际值已事先在子程序声明中规定了。

C 语言程序由程序员定义的若干函数构成。C 不提供类似 PASCAL,ALGOL 语言中过程这种语言单位。然而,可以在 C 的函数体内把控制转回到调用函数处而不回送函数值,这一点很像"过程";当回送函数值时它是一个函数。对于具有回送值的函数,调用程序可以使用这个值,也可以不使用这个值。如果在定义 C 语言函数时,在函数名前冠以 void,则表明该函数无回送值。

C++和 Ada 语言都允许函数名一名多用。即同一个"函数名"可以有不同的含义。

参考文献

1. 徐家福. 系统程序设计语言. 北京:科学出版社,1983

2. 郑国梁,钱士钧. Pascal 语言. 北京:中国铁道出版社,1989

3. 孙玉方,文强. C 语言. 北京:中国铁道出版社,1989

(段祥)

zibianyi chengxu

自编译程序(self-compiler)　用被编译的语言

自身来书写的编译程序。

一般说来,任一编译程序都涉及三个语言:**源语言** S,**目标语言** T 和**实现语言** I。源语言是被该编译程序编译的语言;目标语言是书写目标程序的语言;实现语言是用来书写该编译程序的语言。可以用注明了这三个语言的 T 图(图 1)来表示一个编译程序,或写为 S_IT。例如,用 C 语言写的目标语言为 80386 宏汇编的 C++编译程序可表示为图 2,或写为 $C++_C$ 80386。

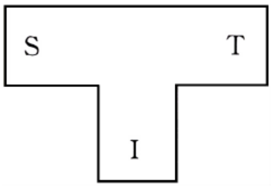


图 1　编译程序的 T 图

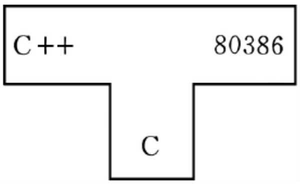


图 2　T 图一例

自编译程序的最大特点是用源语言本身作为实现语言,它的 T 图呈图 3 形式。

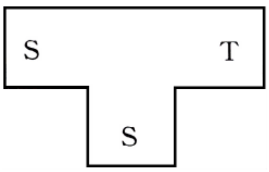


图 3　自编译程序的 T 图

因为语言 S 的编译程序 S_ST 是用 S 写的,因而也是 S 的一个程序,所以该编译程序 S_ST 需要由 S 的另一个编译程序 $S_{I_0}T_0$ 来编译。用 $S_{I_0}T_0$ 对 S_ST 进行编译,得到的结果是用 T_0 写的目标语言为 T 的 S 的编译程序 $S_{T_0}T$。这一编译过程可用如图 4 的 T 图来表示。这种编译过程称为**自展**。

自展有许多用处,例如,可以从一个语言的较小的子集自展出该语言的编译程序;还可以从一个语言的未优化的编译程序构造出该语言的优化编译程序。

参考文献

Aho A V, Sethi R, Ullman J D. Compilers principles, techniques, and tools. Addison-Wesley, 1986

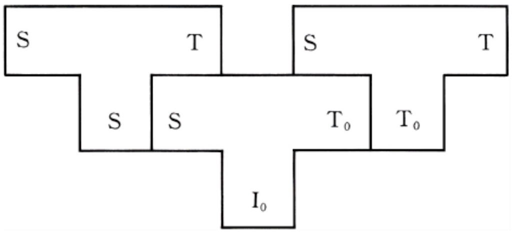


图 4　自展

(徐永森)

zidi xiangshang fangfa

自底向上方法(bottom-up method)　一类软件开发方法,首先从系统实现的最基础部分着手,由简单到复杂,逐层向上构造,直到最后得到所要的软件系统。

以程序设计的情形为例。先实现最基本的子程序、函数,在此基础上构造出功能更强的函数来。每一个新的子程序,或者直接实现,或者调用已经实现的(比它更低层的)子程序来实现。直到最后实现整个系统。方法的特点是:在开发的每一步,系统的开发者手中总是有功能越来越强的子程序可供使用,但是只在最后一步才得到整个系统(也许只需要调用很少几个高层子程序就实现了)。相应的测试工作也可以是自底向上的,即按照构造的同样次序。或者边设计边测试,即对设计好的功能、部件、子程序,立即着手测试。这样,开发者手中随时总是有可信赖的半成品。

代码级的软件复用可以看成是自底向上方法,这时系统的基础部分是可复用构件(参见**软件复用**)。

自底向上方法也可能和其他方法结合,即用其他方法设计(如**自顶向下方法**),而用自底向上方法于实现。在文化程序设计系统中,则可以随意地混合使用各种设计策略,包括自底向上在内(参见**文化程序设计**)。

参考文献

Sodhi J. Software engineering: methods, management, and case tools. Blue Ridge Summit: TAB Professional and Reference Books, 1991

(董韫美)

ziding xiangxia fangfa

自顶向下方法(top-down method)　一类开发

软件的方法,这类方法强调开发过程是由问题到解答、由总体到局部、由一般到具体。自顶向下方法提供了从高层次(问题定义)到低层次(编程实现)的分层分解的决策策略和决策方式,并在具体的开发方法中给出相应的描述工具和设计步骤。自顶向下方法体现了**逐步求精**和信息隐藏的原则。所谓逐步求精即从最能直接反映问题的基本概念和总体结构出发,逐步精化、具体化、补足细节,直到成为可以在机器上执行的程序。逐步求精离不开信息隐藏,在精化的每一层上,把其组成部分的详细内容隐藏起来,留给下层来设计。所以逐步求精使得软件具有较好的层次结构。

发展简史

20 世纪 60 年代中期,在大型软件系统的开发中遇到“软件危机”,促使人们去考察软件设计本身固有的问题。自顶向下方法就是在这一背景下,与结构程序设计及软件工程同时产生的,并一直伴随和促进着软件工程的发展。

1968 年,E. D. Dijkstra 首先指出:为了得到正确的程序,应在编程时采用适当的构造性方法,并指出在程序中使用 GOTO 语句是有害的。他的意见引起了人们对程序设计方法的普遍重视。由此产生了以自顶向下逐步求精为核心内容的结构程序设计方法。

70 年代初,人们对结构程序设计的基本原理及逐步求精方法进行了多方探讨及实际应用,并进行了卓有成效的实践。H. D. Mills 提出了一般意义上的自顶向下开发方法。1970 年 W. Royce 提出了对软件工程具有重要意义的体现自顶向下开发风范的模型——**瀑布模型**,提出了**软件生存周期**的概念,把软件工作涉及的范围划分为需求分析、软件设计、编码测试及运行维护等几个阶段,成为软件开发中行之有效的模型。从此,软件开发方法就不仅局限于编程阶段,而对编程前的分析和设计以及后期的维护等给予了更多的重视。在结构程序设计的基础上提出了结构化分析和结构化设计,形成了结构化开发方法。

E. Yourdon 和 T. DeMarco 等人提出了用数据流程图(DFD)表示系统逻辑结构的结构化分析(SA)方法和用结构图(SC)表示软件模块结构的结构化设计(SD)方法。这种结构化开发方法是以数据流分析和软件的功能分解为特征的。与此同时还产生了侧重于数据结构分析的面向数据结构的一类开发方法。典型的有 K. Orr 和 J. D. Warnier 的**数据结构化系统开发方法**(DSSD)和 M. A. Jackson 的系统开发方法(JSD)以及 D. T. Ross 的**结构化分析设计技术**(SADT)等。

基本内容

自顶向下方法源于结构程序设计,结构程序设计方法的核心就是逐步求精和自顶向下,它的引入从方法学上说是程序设计技术的一场革命,其意义不仅在于当时为解决“软件危机”的实用性一面,而更深远的意义还在于它使程序设计技术成为一种系统的方法,使程序设计从依赖设计者个人的技艺变成一门科学。

逐步求精方法和 Mills 的一般意义上的自顶向下开发方法的基本内容是这样的:首先研究和设计整个系统的结构以及各个子系统之间的关系,并编写和测试总控程序,然后再分析各子系统内部的功能,编写和测试各功能模块,各功能模块分步地联入整个系统中。

结构化分析(SA)方法 SA 方法是在需求分析阶段使用自顶向下、逐层分解的方法,即在顶层抽象地描述整个系统,然后逐层分解,到底层时再具体地描述系统的每个细节。该方法用一套分层的数据流程图表示系统的逻辑模型,用数据字典表示数据特征,用判定表、判定树、结构化语言表示处理的逻辑。分层数据流程图和分层定义数据的方式具体体现了自顶向下逐步求精的过程。

结构化设计(SD)方法 SD 是在结构化分析的基础上,用模块化的方法,自顶向下,根据系统的规约从逻辑模型得出具体的设计方案。该方法的主要工具是描述系统控制结构的模块结构图(SC),它反映软件各组成部分的相互调用关系和相互影响。

数据结构化系统开发(DSSD)方法 这种方法利用顺序、选择和重复三种构造成分来表示分层的信息进而导出软件的结构,使用的基本工具是 Warnier 图。它首先从信息的产生者和接受者的观点,观察数据如何在两者之间流动确定应用环境,然后表达应用功能和给出应用结果。构造 Warnier 图的过程及 Warnier 图的层次结构都体现了自顶向下方法的基本原则。

Jackson 系统开发(JSD)方法 JSD 是在其结构化程序设计(JSP)方法的基础上发展起来的。它以数据结构为基础,通过一组映射或转换来建立软件系统的结构。方法分为实体动作分析、实体结构分析、初始模型定义、功能描述、时间特性的确定和编程实现等几个步骤。

结构化分析与设计技术(SADT) 该方法要求自顶向下地建立 SADT 模型,而模型由一组有序的结构分析图构成,每张结构分析图是由若干个结点以及连结这些结点的弧组成的工程图纸。图有两类,一类称为活动图,一类称为数据图。活动图中的结点表示活动而数据图中的结点表示数据。

总结与展望

自顶向下方法是软件工程和软件方法学的重要内容,这类方法以及与其相应的支持工具已在各类软件的开发中发挥了重要的作用,这类方法基于“把整体划分为相对独立的规模较小的组成部分,比作为整体实现代价小”的软件工程原理,有其独有的优越性。但是这类方法也有它的局限性,自顶向下方法共同存在的缺陷是高层设计对全局影响过大,这就使得如果高层设计得不合理或软件需求发生变化,软件改动的成本太大,甚至要从头设计,带来维护方面的巨大困难。从而软件的重用技术以及提高软件易维护性就成为近代发展的各类自顶向下与自底向上相结合的方法以及面向对象的开发方法所追求的主要目标。

参考文献

Mills H D. Top-down programming in large systems. In: Rustin R, ed. Debugging Techniques in large Systems. New York: Prentice-Hall, 1971. 41-55

(郝克刚　鱼滨)

zidong biaoyin

自动标引(automatic indexing) 使用计算机自动对文献赋予检索标识的过程和技术。

标引(indexing)是指通过对文献的分析,选用确切的检索标识(叙词、关键词等)来反映文献内容的过程。标引工作是信息检索中最重要和最困难的工作之一。标引过程是一项智力活动。传统做法是由标引专家人工完成,但人工标引有一致性差、费用高和效率低的缺点,实现自动标引是实现现代信息检索的重要目标。信息检索领域的自动标引技术同**文本挖掘**或计算语言学领域中的关键词抽取(keyword extraction)及术语自动识别(automatic term recognition)在原理和技术上多有相同之处。

H. P. Luhn 于 1957 年开始基于词频统计的方法进行自动标引研究,其后几十年间,自动标引技术发展得很快,概率统计方法和各种加权模型促进了自动标引技术的发展。20 世纪 90 年代,由于**全文检索**的采用以及**互联网**的兴起,信息需求环境发生变化,自动标引研究渐渐减少。这里**全文检索**是指文献中除了没有检索意义的高频词(通常称停用词或禁用词)以外的每个词都认作标引词。20 世纪 90 年代末至今,随着互联网的发展,很多应用,如自动摘要、文档分类与聚类、文本分析、主题检索等都要依赖于关键词自动提取的结果,自动标引的研究又开始逐渐深入。

根据标引结果来源的不同,自动标引可分为自动抽词标引(也称关键词自动提取)与自动赋词标引两种类型。自动抽词标引即由计算机自动从原文中抽取词或短语作为标引词来描述文献的主题内容。它涉及如何从原文中抽取能够表达其实质意义的词汇,以及如何根据这些词汇确定标引词。自动赋词标引是从预先编制好的某种形式的受控词表中选取词语来表达文献资源的主题内容。即将反映文本主题内容的关键词(欲用作标引的关键词)转换为词表中的主题词(或叙词等),并用其标引的方法。

自动标引技术发展至今的几十年来,研究者对单词标引、短语标引和语义标引等方面都进行了深入研究。

(1)单词标引的目标是找出文献中最主要的单词。单词标引的方法有三种:基于词频统计的标引;基于词区分值的标引;基于概率加权的标引。基于词频统计的方法简单易行,其中逆文献词频对检索效果的改善已被公认。

(2)短语标引的目标是分析文献中“名词 + 名词”或“形容词 + 名词”结构的短语以克服单词标引专指度不够的缺点。一般有两种方法:一种是基于词相关出现频度的统计学方法;另一种是基于句法分析的计算语言学方法。两种方法的查准率基本相同。语言学方法的查全率较高一些,但语言学方法对领域的适应性较差。

(3)语义标引的目的是借助于**自然语言理解**技术,获得对文献的深层理解。由于信息检索系统中领域的无关特性,语义标引有很大困难。主要的研究内容包括省略、指代的处理、布尔逻辑的语义解释、概念和词的语义分类等。一部带有词性、句法、语义和语用知识的词典是语义标引的基础工作。

和英文系统相比较,汉语自动标引还有自动分词的问题,需考虑信息检索用的分词规范和计算语言学中分词规范的差异,例如,在信息检索中,通常把一些词组作为单个检索项,而在计算语言学中,这类词组必须切分为多个词。迄今为止,建立了约十

几个汉语自动标引系统,个别的投入了实用,其基本技术大都基于自动抽词标引,离智能标引还有很大距离。

较为先进的自动标引一般包括如下过程:分词,停用词过滤技术(汉语中的停用词处理较西文复杂),低频词归类处理,高频词组词处理,最优加权函数设计,基于句法分析或词相关频度的短语生成,语言学分析(如指代、省略等),蕴涵概念的推理等。一般认为基于统计学方法和计算语言学方法的结合是实现较高水平自动标引的现实途径。

随着深层语义分析和篇章分析研究的不断深入,这些研究成果可用于自动标引任务,提高标引质量。另外,将多种模型或方法合理地集成,也能在一定程度上提高自动标引的质量。先前的大部分研究工作,主要是针对自动抽词标引的研究,由于资源的匮乏以及词表造价昂贵,自动赋词标引研究与应用相对较少,近年来,随着本体论研究的不断深入,本体有望自动或半自动地被构建,并且可用于自动赋词标引当中。

参考文献

1. Salton G. Automatic text processing: the transformation, analysis, and retrieval of information by computer. Addison Wesley, 1989

2. Frakes W B, Yates R B. Information retrieval: data structures and algorithms. Prentice Hall, 1992

3. 刘挺,秦兵,张宇,车万翔. 信息检索系统导论. 北京:机械工业出版社,2008 (刘挺 施水才)

zidongji lilun

自动机理论(automata theory) 以**离散数学**系统的结构、功能以及两者之间关系为主要研究内容的理论。

自动机可分为以下11种,各有其特点与作用。

有限自动机 一类满足下列条件的离散动态系统:①在考虑的每一时刻系统按输入隶属于有限个可能的状态。②输入从可能状态中选择有限个状态。③任一时刻系统的状态由输入状态和前一时刻系统的状态所确定。电话交换机、升降机、数字电路、神经网络、计算机这类存储量有限的机器都是有限自动机的实例。主要研究(包括开关网络)结构、功能的综合与分析,即讨论功能的数学描述语言(相对应的正规语言)、文法及其性质,状态化简与标准化,状态赋值,各种有限自动机(确定的、不确定的,单向的、双向的,带输出、不带输出的,线性的,自治的,可逆的)的等价性,并设计出满足要求的逻辑网络,有限转换器等。有限自动机已建立了比较完善和系统的研究方法(包括概率、代数、模糊数学方法),是自动机理论的基础。应用于自动控制,生物系统,语言学,通信理论,编码理论,神经网络,计算机高级程序语言的编译设计、词法分析、文本编辑,操作系统,模式识别等。

下推自动机 单向非确定的识别程序,由有限个状态控制一条输入带和一个先进后出的下推表(栈)所组成。有限自动机类是下推自动机类的真子集。主要研究下推自动机的变形和子类(如确定的、双向的、加速的、栈的元素个数的变化等,特别重要的是确定的下推自动机),相对应的上下文无关文法、语言及其运算,不可判定性,下推转换器等。与**有限自动机**一起应用于句法制导翻译模式,比有限自动机更适合于描述各种计算机高级语言。

线性有界自动机 合于条件:①包含输入带左右两端标志的输入字母表。②不能在左(右)端标志向左(右)移动,也不能在左右两端标志处打印任何符号的不确定图灵机。主要研究相应的上下文有关文法、语言及其性质,不可判定问题等。线性有界自动机类真包含下推自动机类且是图灵机类的真子集,还存在着许多公开问题尚待解决。

图灵机 描述通用计算机计算能力的数学模型。主要研究各种构造技术与变形(如无穷输入带的单向、双向扫描、多带、多维、多头、多栈、离线、带外部信息源、状态和符号数受限等)的等价性问题,与**波斯特机**的等价问题,对应的递归可枚举语言性质和文法,停机问题,不可判定问题,丘奇-图灵论题:所有"可计算"函数恰好是部分递归函数,即图灵机可计算函数,通用图灵机等。图灵机可以作为枚举器,是整数函数计算机,在**可计算性理论**和**计算复杂性理论**中有着广泛的应用。图灵机类以线性有界自动机类为其真子集。证实了存在非递归可枚举语言。

时序机 输出与转移函数、转移状态有关的**有限自动机**。主要研究状态的完全化,极小化,时序转移函数的归约与分解,线性时序机,概率时序机,单式前缀、幺半群对正规语言的分解算法与复杂度。应用于研究布尔矩阵的时序开关电路网络的综合与分析,确定冗余性与可靠性,设计计数器等。

波斯特机 由一些基本语句组成的程序框图。它只有一个变量,该变量的取值是字母表 Σ 中字母组成的串(即 Σ^* 的元素)。框图中的语句包括:

①开始语句(只有一个)。②停机语句(接受语句,拒绝语句两个)。③分支判断语句。④赋值语句。Σ^* 的任何元素都可以作为波斯特机的输入,若运行结果在接受(拒绝)语句处停机,表示波斯特机接受(拒绝)该串。Σ 中有一个特殊字母用#表示,它在波斯特对应问题(即同一字母表上任意两个相同长度的字有无匹配的问题)的变形和证明波斯特机与图灵机有同样的计算能力起着重要的作用。波斯特对应问题广泛应用于各种不可判定问题,例如任一上下文无关文法是不是多义这个问题是不可判定的。

随机存取机　由无限多个内装任一整数的存储器 $R_0, R_1, \cdots$ 和有限个内装任一整数的算术寄存器组成,整数均译码为计算机指令。选择适当的指令集编程随机存取机可以模拟现有的任一计算机。提供了基本指令随机存取机与图灵机可相互模拟,是算法分析和**计算复杂性理论**中重要的串行计算模型。主要研究变址形式,从理论上分析计算机串行程序的时、空资源耗费,虚拟并行时间(巡回)实现的变换,判断寄存器、变址器内容是否为 0 实现条件转移等。

栈自动机　满足条件:①输入头为双向,但只读到结束记号。②只读模式时,栈顶压入或弹出动作的同时,栈头可在栈内上下(或左右)扫描但不重写任何符号的**下推自动机**。主要研究各种栈自动机(如确定的,不确定的,可擦符号的,不擦符号的——即永不弹出栈符号,单向的——输入头永不左移,对输出字母表进行考虑,以检验只读可擦输入寄存器是否为空,且栈装有指针,指针可以双向移动,当指针在最下(右)位置时,允许擦去,并打印栈下(右)边的符号,双向的等等),转移表,它所接收、识别的索引语言及其时、空复杂类的特征。应用于推广的上下文无关语言和文章结构。

无限自动机　有限自动机作控制器和存储不受限制,某种数据结构刻画的外部环境组成,是现实计算机的理想化模型。主要研究在计算时、空或其他资源的限制下,自动机所计算的函数类和识别的语言类的描述、包含关系和代数性质、模拟(通过简单的编码,在一个自动机上实现另一个自动机的功能,如图灵机和其他许多自动机可互相模拟)、通用性、不可判定性等。这些都有助于理解计算过程的本质,以及估计算法设计的好坏和固有复杂性下界。

概率自动机　对自动机所处的状态在输入某一字母时下一动作规定条件概率,并给出初始状态的概率分布。动作包括状态的转移,改写字母或输出。主要研究环境或内部具有无限或有限个随机因素的离散数字系统,讨论的参数为可数个,要求电话服务的过程就是这种系统的实例。研究方法一方面推广已有的自动机结果,平行地得到概率图灵机、概率时序机、随机语言类的代数性质(如对某些运算的封闭性),另一方面也考虑了不少新问题,如用 z-变换确定自动机的响应矩阵;估计斯图哈斯蒂矩阵的 k 阶转移应用于多步决策过程;引入自动机的熵和特征值,概率结构的树自动机,应用于模式识别、可靠性问题、动态规划等。

细胞自动机　大量互连**有限自动机**的集合;部件或小计算机互连成大计算机、并行工作部件或计算机的理论模型。将每个自动机想象为一个细胞,由它们构造的子孙自动机至少要有其双亲的功能。20 世纪 50 年代初,冯·诺依曼研究自繁殖复杂机制自动机的逻辑问题时,设计了细胞空间,其中每个细胞都具有相同的领域,不随时间而改变,细胞间的物理距离不影响结果的判断,赋给细胞共振机制并由中央传感器控制,形成庞大、高度分布、同步操作的逻辑成分。根据他提出的概念,发展出许多研究方向,主要有:①迭合空间,②棋盘格空间,③图细胞自动机,④动态细胞自动机——L 系统,⑤Holland 迭代电流计算机(其各细胞有大量状态,邻域可以不同,且随时间可改变),⑥通用计算机构造器(细胞空间及其配置依靠在线交互作用简化)。研究与形式语言的关系,具有一致结构的细胞自动机的综合、分析和容错也是研究的重要内容。应用于超大规模集成电路、并行识别器和并行计算机的设计,模式识别,生物发育,生物化学(开发宏分子和微有机组织,提供研究条件和性质)等。

参考文献

1. 中国大百科全书·电子学与计算机卷. 北京:中国大百科全书出版社,1980

2. Hopcroft J E, Ullman J D. Introduction to automata theory, languages, and computation. Reading, MA: Addison-Wesley, 1979

3. Arbib M A. Theories of abstract automata. Englewood Cliffs, NJ: Prentice-Hall, 1969　(张一立)

zidong jiaohuan guangwangluo

自动交换光网络(automatically switched optical network,ASON)　在光传送网中引入控制

平面,利用信令与路由协议实现网络资源快速动态配置的智能化光网络体系,也称为**智能光网络**。ASON 概念由国际电信联盟标准化部门 ITU-T 于 2000 年所提出。ASON 的本意是以光层网络 OTN 为基础的自动交换传送网 ASTN(automatic switched transport network),但今天 ASON 作为更通用的术语已涵盖了 ASTN 的意义,ASON 与 ASTN 这两个术语许多时候也可混用。ITU-T 的 G.807 建议定义了 ASTN 总体要求,G.8080 建议定义了 ASON 总体结构(见图 1)。

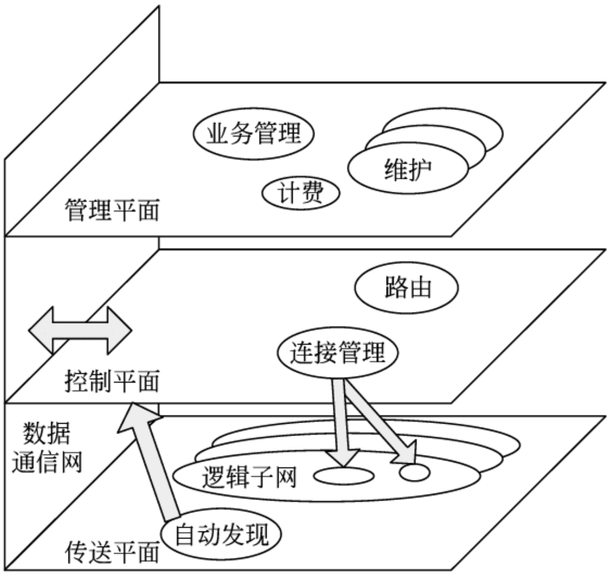


图 1 ASON 体系结构图

ASON 中的三个平面分别完成不同的功能。ASON 传送平面为用户提供从一个端点到另一个端点双向或单向透明的信息传送,同时还要传送控制和网络管理信息。传送平面由传送网元(交叉连接设备和链路)所构成,传送平面中端到端的连接主要是在 ASON 控制平面的控制之下建立,控制平面和管理平面都能对传送网的资源进行操作,这些操作通过传送平面与控制平面和管理平面之间的接口来完成。SDH、OTN 都可以作为传送平面技术,交叉连接设备是 ASON 的核心硬件设备,能组成网状网拓扑,从而有利于寻找最优化的路由或在网络发生故障时快速寻找保护路由,同时也便于在全网共享备份资源。

ASON 中定义的连接类型有三种:①指配型:将沿通道的每一个网元都按请求配置,从而建立端到端连接。指配过程由网管系统或人工干预来完成。这类连接类型称为永久连接(permanent connection,PC)。②信令型:通过控制平面,利用信令消息形式所建立的连接,这类连接称为交换连接(switched connection,SC)。③混合型:由网络提供在网络边缘的永久连接,而利用网络内部的交换连接在网络边缘的永久连接之间提供端到端的连接,这类连接介于交换连接和永久连接之间,也称为软永久连接 SPC(soft permanent connection)。

参考文献

1. 唐雄燕,左鹏. 智能光网络技术与应用实践. 北京:电子工业出版社, 2005

2. 纪越峰,等. 自动交换光网络原理与应用. 北京:北京邮电大学出版社, 2005

(*唐雄燕*)

zidong tuili

自动推理(automated reasoning) 推理过程的计算机实现。尽管自动推理的总体目标是研究机械化推理的各种形式,但它通常被解读为形式逻辑中的合法演绎推理。从这个角度来说,自动推理相当于机器定理证明。构建自动推理程序就是为形式化演算提供一个能在计算机上实现的算法描述,并基于此有效证明该演算下的各种定理。

自动推理的核心问题是:确定一个猜想是否是一组前提假设的逻辑结论。这里的猜想指待验证的属性,而前提假设指研究对象的已知属性。这里的研究对象可以是数学定理、系统、电路、程序、数据类型、通信协议等。我国数学家吴文俊就在几何定理的机械化证明问题上提出了“**吴方法**”,并因此获得自动推理领域的最高奖项——Herbrand 奖。与核心问题相关的,还涉及对猜想和前提假设的**知识表示**,即寻找适当的形式化刻画以表达真实世界,如动作、时间、空间、概念、常识等。经典逻辑是自动推理的主要形式化工具,并且许多证明技术已经被研究和实现。非经典逻辑,如模态逻辑、时序逻辑、**非单调逻辑**、直觉逻辑等也被广泛地用于研究自动推理的知识表示。

我们将推理问题中的前提记为集合 S,猜想记为 φ。一般地,需要寻找由 S 到 φ 的严格证明过程来回答给定的推理问题。这个问题和解决该问题的方法一起构成了定理证明。定理证明包括演绎定理证明和归纳定理证明。其中,演绎定理证明是从 S 推导 φ 的过程,记为 $S \quad \varphi$;归纳定理证明是从 S 推导 φ 的所有基例的过程,记为 $S \quad \varphi^{\sigma}$,其中 σ 为基置换。归纳定理证明的典型代表是数学归纳法。定理证明可以是直接证明,也可以是间接证明。直接证

明是尝试一步一步建立由 S 到 φ 的证明过程,如果证明成功,就表明猜想 φ 成立。典型的方法是德国数学家 Gentzen 提出的自然演绎法。它直接运用经典逻辑推理规则推导结论。间接证明是利用 $S \cup \{\neg\varphi\}$ 生成一个冲突或是不一致,来间接说明 φ 是 S 的逻辑结果。典型的方法有归结法、相继式演算、表演算等。归结法是基于 Herbrand 定理的一阶谓词逻辑反驳法,即把要证明的命题予以否定,再从否定后的命题和条件集出发,通过归结出空子句得到矛盾来证明所证命题为真的推理方法。相继式演算是构造逻辑推导的有效演算。自动推理中常用的表演算方法就是在相继式演算基础上发展的。在定理证明的研究中,对特定情况存在一些特殊处理手段,如相比归结方法更有效判定可满足模型的 DPLL 方法、处理对含有等词的逻辑问题的调解方法、将等式视为规则进行约简的项重写方法等。

自动推理的两个重要性质是可靠性和完备性。可靠性表明演算规则是保真的。对于直接证明,即如果 $S\ \ \varphi$,那么一定有 $S\ \ \varphi$;对于间接证明,即如果 $S \cup \{\neg\varphi\}\ \ \bot$,那么也一定有 $S\ \ \varphi$。完备性则与可靠性相对,它在直接证明中表明,如果 $S\ \ \varphi$,那么一定有 $S\ \ \varphi$;对于间接证明,即如果 $S\ \ \varphi$,那么也一定有 $S \cup \{\neg\varphi\}\ \ \bot$。相比较而言,可靠性较完备性更重要。因为一个演算如果是不完备的,也就意味着某些存在蕴涵关系的推理问题无法通过演算求得;然而如果一个演算是不可靠的,则意味着推理本身是不正确的,是灾难性的。自动推理的另外两个重要性质是判定性和复杂度。一个演算是可判定的,如果它能够在有限时间内对给定问题“$S\ \ \varphi$?”做出是或否的答案。可判定性通常受逻辑、S 对 φ 可接纳的公式形式,以及 S 所表达的理论这三方面影响。可判定并不意味着可实现,因为可判定的推理问题本身就是 NP 完全问题。一个演算的时空复杂度指它的算法效率。自动推理的挑战之一就在于许多演算是不可判定的并且复杂度很高。研究者需要在演算能力和算法效率之间寻找平衡点。

自动推理的研究成果已渗透到各个领域,就人工智能领域而言,自动推理已成为**专家系统**、数据库管理、**机器学习**、机器人、程序验证、**知识工程**的核心。现代自动推理技术的研究不再局限于传统的单调逻辑,非单调逻辑和人类实际推理过程表现的不确定性引起了许多学者的兴趣。近年来,不确定性推理、非单调推理等技术的研究得到迅速发展,在一定程度上反映了人类推理的特点。不确定性推理是指在事实或知识存在不确定性时的推理,一般基于概率、可信度、隶属度等,适用于专家系统问题求解领域。非单调推理是指在推理过程中,在增加某些新事实时,能够取消以前得出的一些结论,更适用于常识推理。但这些推理模型还不能全面反映人类推理的各个方面,因而仍有必要进一步分析人类的推理机制,以便探讨其他形式的推理模型。

参考文献

1. 刘叙华,姜云飞. 定理机器证明. 北京: 科学出版社, 1987

2. Robinson J A, Vornokov A, eds. Handbook of automated reasoning. Elsevier and MIT Press, 2001

(欧阳丹彤　叶育鑫)

zidong wenzhai

自动文摘(automatic summarization)　自动地汇总单篇或多篇文档中的重要信息,以获取一个简洁、连贯、准确反映文档中心内容的摘要的算法及实现技术。早在 1958 年,美国 IBM 公司的 H. P. Luhn 在 IBM704 机器上就开始了自动文摘系统的研制工作,开创了自动文摘的先河。当前摘要任务具有多种不同的具体形式,其中研究中最常涉及的是由迄今国际上最为重要的摘要评测会议 DUC(Document Understanding Conference)和 TAC(Text Analysis Conference)定义的自动摘要任务。从 2008 年起 DUC 的摘要评测任务并入 TAC 评测,其他相关的评测还包括 MSE (Multilingual Summarization Evaluation)、TSC (Text Summarization Challenge), TREC (Text retrieval conference)、TDT (Topic detection and tracking)等会议。这些会议定义了各种自动文摘任务,并给出了较为权威的摘要评测方法和结果。

按照文档数目的不同,文档自动摘要任务可分为单文档文摘和多文档文摘任务;根据是否存在主题查询,分为主题查询无关的自动文摘和主题查询相关的自动文摘任务等;根据文档内容是否动态变化,分为常规文摘和更新文摘(update summarization)等。

多文档自动文摘是近年来的一个研究重点,可以用于海量信息的自动汇总,尤其是**互联网**上针对某一特定事件的信息。当现实世界发生了某个重大事件时,往往会有不同来源的不同方面的报道信息,读者如果想要详细了解发生的事件,则需要阅读大量的相关报道,这就需要花费较多的时间和精力。利用自动文摘技术,就可以将某个事件发生的最重

要信息浓缩在较短的摘要之中，读者仅需通过阅读该摘要就可以了解整个事件的概要，从而大大节省了阅读时间。同样地，自动摘要技术还可用于历史信息的整理、连续事件的追踪等任务之中。例如对事件的持续关注，事件的发生、发展到结束等阶段都包含大量相关信息，自动摘要技术可以从各种信息来源中自动抽取事件的主要内容，并将这些内容按照事件发展的顺序组织起来，使读者可以通过阅读摘要就迅速了解整个事件的轮廓。基于主题查询的多文档文摘则进一步考虑用户的查询要求，旨在从特定事件相关的大量文档中自动生成与用户查询相关的摘要。搜索引擎中的查询通常为词语或短语的组合，这里的用户查询一般用一个句子表达，通常表现为几个较长的疑问句。

随着**自然语言处理**、**机器学习**等技术的不断进步，传统的基于文档检索模型的搜索引擎不再限于简单的关键词查询，自动文摘技术可以对搜索引擎返回的信息进行精炼、提取，然后返回给用户，从而使呈现给用户的结果更为人性化。查询相关的多文档摘要模拟的正是在搜索引擎返回相关文档后，从文档中抽取出满足用户查询信息的过程。可以预见，自动摘要将会成为构成下一代智能搜索引擎的关键技术之一。

参考文献

1. Luhn H P. The automatic creation of literature abstract. IBM Journal of Research and Development，1958，2(2)：159-165

2. 秦兵，刘挺，李生. 多文档自动文摘综述. 中文信息学报，2005，19(6)：13-20

（李素建）

zijianyan dianlu

自检验电路(self-checking circuits)　不需要外加测试码而能自动地证实电路本身或系统中是否存在故障的电路。又称自校验电路或者自检电路，如图1所示。自检验电路的输出进行了错误检测编码，一旦输出信号出现了非编码字节，监控器能立即检测到错误。

自检验电路中的内测机制(built-in self-test mechanism，BIST)可以分为并发和非并发两类。并发BIST在进行正常操作的同时进行在线同步检测；非并发BIST在正常操作前，自动进行了离线检测。其中，具有并发错误检测(concurrent error detection，CED)机制的电路能检测出瞬时和永久性故障，被广泛用于对可靠性和数据完整性要求较高的系统中。

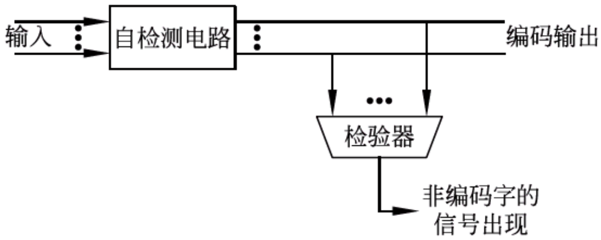


图1　自检验电路检测示意图

CED技术同时也提高了离线易测性，减少了BIST的负载。所有的CED电路分为两个模块：功能逻辑模块和检测器模块。功能逻辑模块对输出信号进行错误检测编码，检测器模块判断输出信号是否为一个编码字。

假设电路有p个输入端和q个输出端，则可能出现的输入组合为2^p个，称为输入空间，可能出现的输出组合为2^q个，称为输出空间。如果在正常运行时输出空间中的2^q个组合均可能出现，则就无法通过观察和验证输出值来确定电路是否存在故障。如果在输出空间2^q中仅有$k<2^q$个输出组合为有效的，那么在出现另一些属于2^q-k中的组合时就可断定电路中必定存在故障。通常属于k中的组合为有效的输出组合，称为合法码字；否则，即为错误的输出组合，称为非法码字。

若电路中任一故障f出现时，都能使该电路的输出成为非法码字，则称该电路为故障安全电路。若对电路中的任一故障至少存在1个输入组合使得该故障发生时其相应的输出组合为非法码字，则称该电路是自测试的。若一个电路既是故障安全，又是自测试的，则称该电路为完全自检验电路。

在自校验电路中，通常设置1个单输出的校验器。当电路正常时，校验器输出为0；在发生故障时，校验器输出为1。这样就起到了校验电路的作用。但是这时也必须注意，校验器本身的输出端是否存在固定1的故障。因为当校验器输出发生固定1的故障后，就会对被校验电路产生错检或漏检现象。

自检验电路通常具有以下两个特点：自检测和故障安全。分别解释如下：

(1)自检测　对于预先定义集合中所产生的每个错误，电路都能为至少一个编码空间输入产生一个非编码空间的输出，称这种电路是自检测的。

(2)故障安全　对于预先定义集合中所产生的每个错误，电路不会对编码空间输入产生错误的编码空间输出，称这种电路是故障安全的。这里定义

中的代码空间是指正常输入的编码。

参考文献

1. Dubrova E. A design technique for high-performance self-checking combinational circuits. Proceedings of MAPLD'01, 2001: 11-13

2. Sekanina L. Evolution of polymorphic self-checking circuits. Proceedings of ICES'07, 2007: 186-197

（谢夏　徐拾义）

ziran jingwu zaoxing

自然景物造型（modeling of natural phenomena）　在计算机系统中生成自然界景物模型的过程、方法和技术。许多复杂的自然景物，如天空、山脉、河流、海洋、大地、树木、花草、云、烟、雾等，其几何形貌常常既有一定的规则性又有很大的随机性。由于其形状的不规则性，很难用欧氏空间的连续函数来描述，但可以通过一些能近似反映其形状规律的过程式方法来模拟。这类造型技术的主要特点是能够逼真地再现自然景象，而不要求模型与实际景物完全一致。常用的方法有分形迭代造型技术、基于文法规则的造型技术、动态过程造型技术、基于纹理实例的造型技术和基于物理模型的造型技术等。

分形迭代造型技术是用递归方法来模拟具有自相似性（从严格意义上或从统计意义上）的、随机的复杂形体的一种造型技术，如海岸线、山峦、花、云等。这种方法最早出现于 1904 年，19 世纪 60 年代 B. B. Mandelbrot 正式提出了分数维的概念，Hutchinson 于 1981 提出迭代函数系统 IFS 模拟植物。

基于文法规则的造型技术采用文法生成结构性较强的形体拓扑结构（如树木），再通过几何解释来形成形体的图形（如树干、树叶），包括 L-系统、A-系统和植物生长模型。1968 年，Lindenmayer 等提出了描述植物形态与生长的 L-系统，1971 年，Honda 提出了构造植物形态的 A-系统。

动态过程造型技术是一种表示景物的动态过程变化模型的造型方法，包括粒子系统、细胞自动机方法等。粒子系统采用大量随时间变化的粒子图元来描述动态景物，随着时间变化对粒子状态进行更新，粒子可以产生和消亡，在粒子运动过程中遵循确定的或随机的运动规律。最早的粒子系统是由 W. T. Teeves 于 1983 年提出来的。细胞自动机是通过一个局部可预测的规则来获得离散动力学系统模型，1991 年 Pakeshi 等提出细胞自动机的火焰模型。

基于纹理实例的造型技术是利用采集的物体纹理样本来合成的造型方法。利用树木草地等景物在形态和结构上的相似性，构造有限个体纹理样本，再对纹理的大小、形状和方向引入一定的随机性来合成大尺度景物，包括纹元、体纹理、视频纹理等。1989 年，Kajiya 和 Kay 提出纹元的概念。

基于物理模型的造型技术是从自然现象的物理原理出发，进行面向仿真的适当简化，然后进行建模和绘制的方法。对应于不同现象的物理模型，可以采用相应的简化加速方法。如基于流体方程的龙卷风、泥石流等流体现象的模拟，基于大气折射散射模型的天空现象、蜃景、宝光及彩虹的模拟，基于动力学运动模型的风吹草动、森林的模拟等。

参考文献

1. Foley J D, van Dam A, et al. Computer graphics: principles and practice. 2nd ed. Addison-Wesley, 1995

2. 彭群生，鲍虎军，金小刚. 计算机真实感图形的算法基础. 北京：科学出版社，1999　（王长波）

ziran yanyi fa

自然演绎法（natural deduction method）　机器证明定理领域一种重要的证明方法，也称为自然演绎或自然推导法，由 W. W. Bledsoe 于 1975 年提出。

在 1956 年机器证明定理研究领域出现以后的二十多年的时间里，机器证明定理领域的研究者主要使用**归结方法**。这种方法的理论基础非常完善，只使用一条规则——归结规则，技术实现上比较容易。而且是半可判定的：对**一阶逻辑**中的任一个公式，只要它是恒真的，都可以用归结方法给出证明。但是归结方法也有自身的弱点——效率很低。其原因是在证明过程中会产生大量归结式，证明过程常常会因为归结式数量太多耗尽了计算机的资源而得不到最终结果。定理机器证明领域后来的研究者的主要工作是研究限制归结式大量产生的方法，提高归结方法的效率，但一直没有很大的突破。

1975 年，为提高机器证明定理的效率，Bledsoe 提出一种在理论上与归结方法完全不同的证明方法，这种方法模拟人的演绎思维过程，把推理的依据转换成一些有效的启发式规则。利用这些规则实现推理过程。取得了较好效果。1966 年美国麻省理工学院的 L. Norton 建造了一个系统 ADEPT，该系统是一个关于群论的启发式定理证明系统，它使用

了很多有效的启发式规则。虽然这个系统的能力有限,但它是第一个模拟人的演绎思维过程并在计算机上实现的定理证明方法。

自然演绎法与归结方法在很多方面有本质的区别。首先,自然演绎法证明的主导思想与归结方法不同。假设我们要证明的公式是 $H \Rightarrow C$,其中 $H = \{h_1, h_2, \cdots, h_n\}$ 是前提集合,C 是结论集合,归结方法实际上采用的是反驳证明方法,转去证明 $H \wedge (\sim C)$ 的不可满足性。而自然演绎法却是直接证明 $H \Rightarrow C$ 的恒真性,其演绎过程与人们证明这个问题的思维过程十分类似。自然演绎法在实施具体推导过程之前,需要把 H 和 C 中的一阶逻辑公式转换成推理规则,也需要使用 Skolem 化过程。因为证明的主导思想不同,自然演绎法在对结论公式 C 的 Skolem 化过程与归结方法不同。归结方法的 Skolem 化的结果是全称量化的,存在量词用 Skolem 函数代替,而自然演绎法恰恰相反,Skolem 化的结果是存在量化的,全称量词用 Skolem 函数代替。归结方法从反证角度出发,对前提条件和要证明的结论不加区别。而自然演绎法却在证明中始终保持它们的区别,不断地分析前提和结论的逻辑结构,根据它们的结构选用不同的演绎规则。自然演绎法的优点是演绎过程中所产生的子目标数量较少,因而占用的存储空间较少,证明问题的速度比较快。中等难度的问题在人们容许的时间内即可得到证明。由于它的演绎过程和人对问题的证明过程类似,因而便于人机对话,实现交互式证明。例如,如果机器证明时间过长或者在证明过程中遇见问题时,可以中断证明,请求人的帮助与指导。

自然演绎法的弱点是它的不完备性——有些人利用通常的推理方法能够证明的定理,使用自然演绎法却无法证明。为了解决自然演绎法的不完备性问题,人们对自然演绎规则提出了各种修改意见。这些修改虽然能够解决自然演绎法在某些具体问题上的不完备性,却不能从根本上彻底解决它的不完备性。自然演绎法的不完备性之有效解决,至今仍是一个大问题。

参考文献

Bledsoe W W. Non-resolution theorem proving. Artificial Intelligence, 1977, 9(1): 1-35 (姜云飞)

ziran yonghu jiemian

自然用户界面(natural user interface) 利用用户自然习得的知识,在自然交互环境中通过自然交互方式实现人机交互的用户界面。

用户界面的发展包括命令行、图形用户界面和正在兴起的自然用户界面三个阶段。在命令行时代,用户必须按照严格的语法规则通过键盘输入命令,只能获得非常有限的反馈。图形用户界面利用了桌面隐喻,用户主要通过鼠标的运动和点击,与图形化方式呈现的界面进行交互。图形用户界面的主要界面范式是 WIMP(window, icon, menu, pointing device),但是 WIMP 界面在多个方面都有内在的缺陷,只能支持单通道、离散和精确的交互方式。Mark Green 和 Robert Jacob 在 1991 年提出了 non-WIMP 用户界面,Andries van Dam 在 1997 年提出了 post-WIMP 用户界面。今天,随着软硬件技术的飞速发展,计算设备逐步融入人们日常的生活和工作环境中,并能够采集、融合、理解用户交互信息和交互意图,从而使得用户能够在自然的交互环境中以自然的方式进行交互。

自然用户界面是对传统用户界面的彻底改变。首先,用户进行交互所需的知识不再是计算机的规则和命令,而是自己自然习得的知识和经验,极大减少了用户的学习时间,使用户能够快速完成从新手用户到专家用户的迁移;其次,用户的交互环境不再局限于传统基于桌面隐喻的办公环境,而是扩大到日常生活和工作的各个领域,用户可以随时随地与计算环境进行交互;最后,用户的交互方式也不再局限于鼠标的指点和点击,而是可以通过动作、运动、手势、言语,甚至生理信号、脑电信号等单通道或多通道融合的自然方式进行交互,拓展了交互通道和交互能力。

目前,自然用户界面的研究刚刚起步。人机交互领域的研究者仅在特定的领域中开展了相关研究,并获得了阶段性成果。这些研究领域包括:普适计算、上下文感知计算、情感计算、**虚拟现实**、混合和增强现实、移动交互、体感控制、语音和多通道交互、隐式交互等研究方面。但是并没有建立起自然用户界面相应的理论体系和技术框架,缺乏对自然用户界面统一的评估模型和衡量指标。未来若干年的自然用户界面的研究热点将集中在自然用户界面的基础理论、评估框架和关键技术方面。

参考文献

1. Jacob R J K, et al. Reality-based interaction: a framework for post-WIMP interfaces. ACM SICHI 2008 Conference, 2008

2. 戴国忠,笔式用户界面. 合肥:中国科学技

术大学出版社,2009 （戴国忠）

ziran yuyan chuli

自然语言处理（natural language processing，NLP） 利用计算机技术对自然语言文本(句子、篇章或话语等)进行处理和加工的一门学科。自然语言是指人类社会发展过程中自然产生的语言，如汉语、英语等，通常有书面语和口语两种形式，不包括计算机程序语言。NLP 的研究内容包括对词法、句法、语义和语用等信息的识别、分类、提取、转换和生成等各种处理方法和实现技术。

自然语言处理起源于早期**人工智能**对**自然语言理解**(natural language understanding，NLU)问题的研究，但在人工智能研究中更强调对人类自身语言能力和思维本质的探索，希望这项研究有助于制造出能够模仿人类高级智能行为的机器。因此，从一开始它便是人工智能学科的一个重要分支，是在语言学、计算机科学、认知科学、信息论和数学等多学科基础上形成的一门交叉学科。

自然语言处理技术可以广泛地应用于**问答系统**(question-answering system)、**自动文摘**(automatic summarization)、**机器翻译**(machine translation)、文本释义(paraphrase)、**文本挖掘**(text mining)、**信息提取**(information extraction)、文本分类(text categorization)、情感分类(sentiment classification)和隐喻计算(metaphorical computation)等。

自然语言处理技术的发展大体上经历了三个阶段，即 20 世纪 60 年代及更早以关键词匹配为主流技术的早期，70 年代至 80 年代以句法-语义分析方法为主流的中期和 90 年代后以语料库和统计语言模型为主流技术的快速发展时期。早期的自然语言处理系统大多没有真正的语法分析，而主要依赖关键词匹配技术来识别输入句子的含义；进入 70 年代以后，一批采用句法-语义分析方法的自然语言处理系统脱颖而出，它们在语言分析的难度和深度方面都比早期系统有了长足进步。这些进步很大程度上得益于**计算语言学**在语言理论方面的成果；自 20 世纪 90 年代以来，大规模语料库和基于语料库的统计语言模型首先在语音识别、机器翻译和信息检索(搜索引擎)等应用系统中获得成功，并被迅速推广到句法分析、信息抽取、自动文摘、文本挖掘等自然语言处理系统的研究与开发中。这种方法的最大特点是把一个自然语言处理的任务看成是**机器学习**问题，因而若干机器学习方法和数学模型在这一领域发挥了重要作用。因此，在某种意义上“自然语言处理”带有一定的“语言工程”色彩。

随着**互联网**和通信技术的迅速发展和普及，自然语言处理技术正在更多的应用需求驱使下，向着分析方法与统计方法相融合的方向，并伴随着高性能计算能力的提高和认知科学的进展而快速发展。

参考文献

1. 石纯一，黄昌宁，王家廞. 人工智能原理. 北京：清华大学出版社，1993

2. 冯志伟. 自然语言的计算机处理. 上海：上海外语教育出版社，1996

3. 宗成庆. 统计自然语言处理. 2 版. 北京：清华大学出版社，2013

（宗成庆　黄昌宁）

ziran yuyan fenxi

自然语言分析（natural language analysis） 计算机根据给定的形式化的语言知识模型，将自然语言文本中的词语单元、句法结构、语义关系、语篇框架等信息转换为特定的计算机机内表示的技术。运用自然语言分析技术对自然语言文本加以分析，是进一步做信息检索、**信息提取**、文本摘要、**机器翻译**等各种自然语言信息处理的基础。通常可根据分析的自然语言单位大小不同以及分析的目标不同，把自然语言分析任务分为词法分析、句法分析、语义分析、语篇分析等子任务。

词法分析包括连续字符流中的词语识别，词语内部结构分析和形态分析，词类标注等具体分析任务。显然，词法分析跟所处理的具体语言有关，对于有词形屈折变化的语言，如英语、德语等，需要进行形态分析。对于没有形态变化的语言，如汉语，则不需要进行一般的形态分析。不过汉语也有自己的特点，需要做特殊的词法分析，例如汉语在书写时词和词之间没有空格。进行汉语书面文本的信息处理，首先就需要将汉语文本中的词逐个识别出来，在词和词之间加上分界标记。这个过程通常称作汉语词语切分。在这个过程中需要处理汉语的离合词、重叠词等特殊形式，也可看作是广义的词语形态分析。词法分析中的词类标注任务是确定文本中词的词类信息，这可以为后续的词义分析及句子结构分析提供帮助。自然语言中一词多类的现象比较常见。当兼类词出现在文本中时，需要根据语境来动态确定其词类。

句法分析处理的基本单位是句子，主要目标是判断输入句子是否是自然语言中合乎语法的表达

式。如果是合语法的,则应分析得到能够反映句子组成情况的结构,通常用树形图表示。常见的有短语结构树和依存结构树两种形式。除对整句的句法结构进行完全句法分析外,也有所谓的浅层句法分析,即对句中特定的语言单位进行识别,比如识别出句中的命名实体、时间、处所表达式等等,因为这种分析只给出一个句子中的部分成分作为结果,所以又称部分句法分析。

语义分析主要有两类任务：一是词义分析,处理的单位是词;二是句义分析,处理的单位是句子。词义分析又称作词义标注。自然语言中的多义词现象比较常见,这些词在不同的语境中可以解释为不同的义项,词义分析的目标就是确定文本中的多义词在当前语境中所表现的义项。句义分析的目标是得到句子意义的形式化表示,通常在句法分析之后进行。典型的做法是按照一定的语义规则在句法分析的基础上推导出句子的意义表示。一般可以表示为谓词逻辑表达式,特征结构,语义网络等形式。除对于整句的完全语义分析外,也有所谓的浅层句义分析,一般又称作语义角色标注,分析目标是识别句子中心谓词的论元成分所充当的不同语义角色。典型的语义角色有施事(动作者)、受事(受动者)、与事(参与者)、工具、材料、时间、处所等。

语篇分析的对象是由句子组成的段落和完整的篇章。分析任务主要有两类：一是微观层次上的篇章分析,即确定篇章中实体成分之间的指代关系。主要是代词与其所指代的语言成分之间的共指关系。二是宏观层次上的篇章分析,即分析篇章的整体结构以及句间逻辑语义关系,如因果、假设、条件、让步、转折等关系。

上述各级语言单位的分析在具体实现时大致都可以区分为基于规则的方法和基于统计的方法两种模式。就自然语言处理的发展历史而言,早期的研究范式以基于规则的方法为主,近期则转为以基于统计的方法为主。基于规则方法的一般工作模式是,由人工提出一个语言模型,比如以上下文无关语法形式表达的语言成分组合规则,然后设计相应的搜索算法,对输入字符串进行扫描,搜索规则集中的规则,得到在模型意义下可以解释输入字符串的合理结构。在这个工作模式下,语言模型对语言知识的刻画一般是遵循布尔逻辑的原则,即语言表达式要么是符合规则的(取值为 1),要么是不符合规则的(取值为 0)。基于统计方法的一般工作模式则是,根据人工提出的对语言单位的初始分类,对大规模语言实例进行标注形成语料库,然后建立特定的数学统计模型(比如隐马尔可夫模型、最大熵模型、条件随机场模型),从语料库中获取语言模型的参数。这个过程称为参数训练。根据训练所得参数来计算输入字符串的各种可能的结构在模型意义下的最大概率,从而选择一个最优解。这个过程一般又称为解码。在这个工作模式下,语言模型对语言知识的刻画是遵循概率分布的原则,即语言表达式之间的差异体现为在[0,1]区间内取值的分布概率,而不再仅仅是 0 跟 1 的二值对立。基于规则方法的哲学基础是所谓的理性主义,基于统计方法的哲学基础则是所谓的经验主义。尽管目前还很难说哪一种方法在自然语言处理问题上更有效,但由于自然语言客观存在的极大复杂性,以人工规则方式刻画语言知识的粒度一般来说比以统计方式刻画语言知识的粒度要粗。近年来随着统计机器学习研究的迅速发展,基于统计的方法,或者说由数据驱动的分析方法在自然语言分析技术领域得到了更为广泛的重视和应用。

除词法、句法、语义、篇章分析等传统的自然语言分析任务外,近年来还因互联网信息检索和信息抽取等应用的发展需要,产生了新的分析任务：**情感分析**和**隐喻分析**。文本情感分析通常按照处理对象的不同,分为 4 个层次,包括：①词语情感倾向性分析;②句子情感倾向分析;③篇章情感倾向性分析;④超大文本整体倾向性预测。其中每个层次的研究又可区分出不同的具体任务,比如词语情感倾向性分析就包括 3 个具体的任务：情感词的识别;情感词褒贬义的区分;情感词倾向义程度的度量。隐喻分析又称隐喻识别,即发现文本中的隐喻表达。目前的主要方法有基于文本线索的方法、基于语义知识的方法和基于机器学习的方法等。

参考文献

1. Mitkov R, ed. The Oxford handbook of computational linguistics. Oxford University Press, 2003

2. 俞士汶. 计算语言学概论. 北京：商务印书馆,2003

(詹卫东)

ziran yuyan lijie

自然语言理解(natural language understanding,NLU) 探索人类自身语言能力和语言思维活动的本质,研究模仿人类语言认知过程的**自然语言处理**方法和实现技术的一门学科。它是**人工智能**早期研究的领域之一,是一门在语言学、计算机

科学、认知科学、信息论和数学等多学科基础上形成的交叉学科。

相对于自然语言处理，自然语言理解更强调对人类自身语言认知过程和思维活动本质的研究和探索，以及模拟人类思维建立计算机处理和加工自然语言句子、篇章或话语等信息的方法和手段。

对于什么是“理解”目前并没有统一的解释。但在人工智能界，有专家认为可以采用图灵试验来判断计算机是否“理解”了自然语言，具体的判据分述如下：

（1）问答　机器能正确摘取输入文本中的主要信息，并据此回答有关的问题；

（2）释义　机器能用不同的词语和句型来复述输入文本；

（3）文摘生成　机器有能力产生输入文本的摘要；

（4）翻译　机器具有把一种源语言文本翻译成另一种指定的目标语言的能力。

关于自然语言理解的研究大致可以归纳为两个方向：一是以探索人类大脑语言理解过程为目标的认知方法研究；二是以建立自然语言处理系统为目标的实用技术和方法研究。在语言认知过程的研究中，人们通常聚集于揭示人脑在语言思维活动中的规律，通过认知的、心理的、甚至医学的实验手段，发现和解释人脑在语言理解过程中的各种现象。这类研究与认知科学和语言心理学有密切的联系；而在以建立应用系统为目标的研究中，则更加关注于研究语言知识的形式化表示和处理算法及模型等问题，通过建立**语言知识库**、语料库和数学模型等手段实现特定任务的自然语言处理系统。这类研究往往与语言学、计算机科学和数学密切相关。通常情况下，我们所说的自然语言理解较多地倾向于后者。从这一角度讲，自然语言理解研究大致经历了三个不同的发展时期，即20世纪60年代及更早以关键词匹配为主流技术的萌芽时期；70年代至80年代以理性主义方法为主导的、以句法-语义分析方法为主流的发展时期；90年代之后从经验主义复兴、以语料库和统计语言模型为主流技术到经验主义方法与理性主义方法相融合的快速发展时期。

随着互联网和通信技术的迅速发展和普及，自然语言理解技术在**问答系统**（question-answering system）、**自动文摘**（automatic summarization）、**机器翻译**（machine translation）和**信息提取**（information extraction）等具有广阔的应用前景。

参考文献

1. 石纯一，黄昌宁，等. 人工智能原理. 北京：清华大学出版社，1993

2. 翁富良，王野翊. 计算语言学导论. 北京：中国社会科学出版社，1998

3. 冯志伟. 计算语言学基础. 北京：商务印书馆，2001

（宗成庆　黄昌宁）

ziran yuyan shengcheng

自然语言生成（natural language generation）　计算机根据给定的形式化的语言知识模型，将特定表达意图的机内形式化表示转换为自然语言形式输出的技术。自然语言生成技术可应用于**机器翻译**、**自动文摘**、多语信息发布等**自然语言处理**系统。自然语言生成过程一般视作一种由交际目标驱动的规划过程。通常按照生成的自然语言单位从大到小的顺序，把自然语言生成任务分解为文档规划、句子规划、表层实现等子任务。

文档规划　主要任务是确定文本的内容以及文档的整体结构，其目标是生成一个关于待生成文本的规格说明，即决定文本中应该传达哪些信息。影响文本内容确定的因素主要有文本的交际意图、文本预期读者的信息、对输出文本的要求以及文本的信息来源等。文本不是材料的任意堆积，在文本信息内容确定后，文档规划还需要就这些信息的排序和结构作出决策，完成这一任务的模块通常称作文档结构化模块。文本结构通常是一个树形结构，规定了信息如何分组以及信息分组之间的语篇关系。

句子规划　在文档规划的基础上进行，主要完成三方面的工作：一是词汇选择，选择适于表达文本信息的动词、名词、形容词等实词性词汇；二是指称表达的生成，生成文本涉及的人、时、地等实体性元素的具体表达形式，如限定式名词短语，代词等；三是整合，完成文档规划阶段所生成的文本结构树到篇、段、句等文本元素的映射。为了生成自然流畅的文本，文本整合模块需要对内容类似的句子进行归并，避免生成冗长刻板的表达。

表层实现　任务可分为两个方面，一是语言实现，即在句子规划阶段所确定的抽象的语篇元素的基础上，按照具体语言的形态、句法规则将其实现为合法的自然语言句子乃至完整语篇。例如对句子中使用哪些虚词作出选择；确定句子的时态、名词的性、数、格标记形式，等等；二是文本的格式化，按照

具体生成系统的要求,对最终生成的文本元素进行格式标记,满足系统输出格式的需求。

参考文献

1. Mitkov R, ed. The Oxford handbook of computational linguistics. Oxford University Press, 2003

2. Reiter E, Dale R. Building natural language generation systems. Cambridge University Press, 2000

3. 俞士汶. 计算语言学概论. 北京: 商务印书馆, 2003

（詹卫东）

zishiying xiezhen lilun

自适应谐振理论(adaptive resonance theory, ART)　一种能自组织地产生对环境认知编码的**人工神经网络**理论模型,简称 ART 模型。神经网络通过和外部环境的交互作用,在神经网络中自发地产生对环境信息的编码,进行自组织活动,是一种无监督学习的网络。

ART 模型是美国 Boston 大学的 S. Grossberg 和 A. Carpenet 在 1976 年提出的,它来源于 Helmboltz 无意识推理学说的协作-竞争网络交互模型。Grossberg 一直对人类的心理和认识活动感兴趣,他长期埋头于这方面的研究并希望用数学来刻划人类这项活动,建立人类的心理和认知活动的一种统一的数学模型和理论。ART 就是由这种理论的核心内容并经过提高发展然后得出的。

ART 模型的框图如图 1 所示。

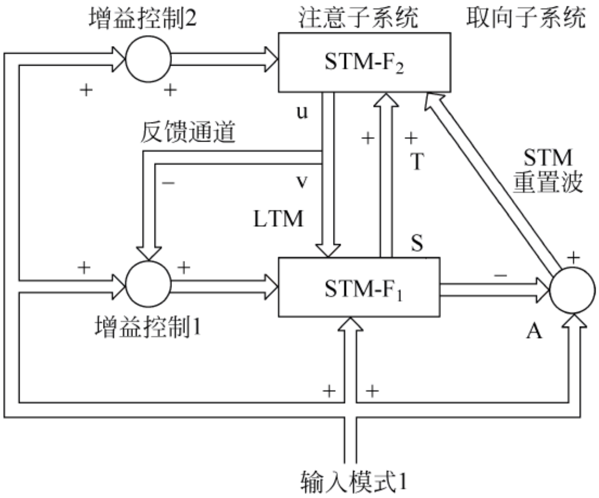


图 1　ART 模型框图

有两个短时记忆层 STM-F1 和 STM-F2,通过长时记忆层 LTM 连接。它由两个子系统组成,一个称为注意子系统(Attentional Subsystem),一个称为取向子系统(Orienting Subsystem),也称调整子系统。这两个子系统是功能互补的子系统。ART 模型就是通过这两个子系统和控制机制之间的交互作用来处理熟悉的事件或不熟悉的事件。增益控制有两个作用:一个作用是在短时记忆层 STM-F1,用于区别自下而上和自上而下的信号;另一作用是当输入信号进入系统时,短时记忆层 STM-F2 能够对来自 STM-F1 的信号起阈值作用。调整子系统是由调整节点 A 和短时记忆 STM 重置波通道组成。注意子系统的作用是对熟悉事件进行处理。调整子系统的作用是对不熟悉事件产生响应。ART 模型就是由注意子系统和调整子系统共同作用,从而完成自组织过程的。

Grossberg 所提出的 ART 模型有如下一些主要优点:①可以进行实时学习,能适应非平稳的环境。②对于已经学习过的对象具有稳定的快速识别能力;同时,亦能迅速适应未学习的新对象。③具有自归一能力,根据某些特征在全体中所占的比例,有时作为关键特征,有时当作噪声处理。④不需要预先知道样本结果,是无监督学习;如果对环境作出错误反映则自动提高"警觉性",迅速识别对象。⑤容量不受输入通道数的限制,存储对象也不要是正交的。

ART 模型已有 ART1, ART2, ART3, Fuzzy ART 等几种模型结构。ART1 用于处理二进制输入的信息;ART2 用于处理二进制和模拟信息这两种输入;ART3 用于进行分级搜索;Fuzzy ART 在模式分类中使用模糊逻辑,因此提高了泛化能力。目前已经发现 Fuzzy ART 和 ART1 的训练结果严重依赖于处理数据的输入顺序。这种缺点虽然可以通过使用较小的学习率得到一定程度的缓解,但增加输入样本的数目却不能消除这种影响。因此, Fuzzy ART 和 ART1 的估计不具有统计意义上的一致性。

ART 模型可以用于语音、视觉、嗅觉和字符识别等领域。最近,人们基于 ART 模型的机理,研究类脑智能机。

参考文献

1. 史忠植. 神经计算. 北京: 电子工业出版社,1993

2. Carpenter G A, Grossberg S. Adaptive resonance theory, In: Michael A. Arbib (eds.) The handbook of brain theory and neural networks. 2nd ed. Cambridge, MA: MIT Press,2003

（叶世伟）

zizuzhi yingshe

自组织映射(self organizing maps,SOM)

由输入层和输出层组成的两层式神经网络。通过设定的一种侧向抑制的简单竞争算法,实现类似于感觉信号映射到大脑皮层的有序特征映射,从而实现对输入样本的排序和分类。

神经生理学家很早就发现,厚约 2 mm,总表面积约 15 m^2的大脑皮层是按功能分区的。例如有运动皮层、体感皮层和初级视觉皮层等。在听觉皮层区域内,神经元的排列反映出频率的对数刻度,低频信号引起该皮层一端区域的神经元反应,而高频信号则引起另一端的神经元反应。这表明,感觉信号的各种特征,是按照其取值的大小,有组织地排列在大脑皮层的一定空间范围内的。当人脑通过感觉器官接受外界的特定信息时,大脑皮层的相应区域就兴奋,并且类似的外界信息映射到相应的一个区域。这种响应特性并不是先天赋有的,而是通过后天学习经过自组织形成的。具体来说,各个神经元都和相邻神经元经过侧向连接相互作用,通过这种作用(竞争学习)可以自适应地发展成为对不同性质信号敏感的区域。

20 世纪 80 年代芬兰赫尔辛基大学信息科学系 T. Kohonen 教授提出的自组织映射方法,就是用**人工神经网络**方法来实现这种输入样本到输出二维平面的有序特征映射。图 1 表示出这种自组织映射网络。由图可见,自组织映射网络有二层,输入层模拟感知外界信息的器官(例如视网膜),输出层模拟给出响应的大脑皮层。它的学习算法主要由竞争、协作和自适应调整三个过程构成,具体步骤如下:

(1) 权值的初始化,用小的随机数对各连接的权赋不相同初值 $w_{ij}(0)$。

(2) 在样本集中任选一个样本 $\boldsymbol{x}$ 作为输入,$\boldsymbol{x}$ 是 d 维向量,其中 d 为输入单元个数。

(3) 在时刻 n 选择输入层中的最佳匹配单元(竞争过程),即:

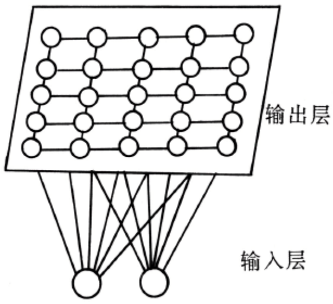
$i(x)=\underset{j}{\operatorname{argmin}}\|x(n)-w_j\|$, $j=1,2,\cdots,N$. i 为获胜单元。

(4) 确定邻域函数(协作过程)$\Lambda_i(n)$。

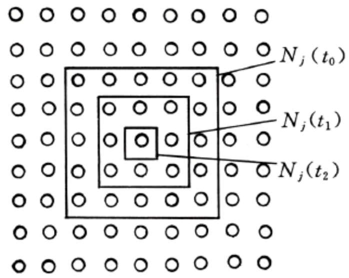
(5) 修正权值(自适应调整过程)。

$$w_j(n+1)=\begin{cases} w_j(n)+\eta(n)[x(n)-w_j(n)] & j\in\Lambda(n) \\ w_j(n) & \text{其他} \end{cases}$$

(6) 转步骤 2,依次输入其余样本。对上述过程重复直到到达给定的学习次数。



(a) 网络拓扑结构



(b) $N_j(t)$形状变化

图 1 自组织映射网络

在学习过程中随时间缩小邻域函数 $\Lambda(n)$ 并减小学习步长 $\eta(n)$,直到满足事先给定的终止条件(例如相邻两次学习中权值的变化小于给定值)。

自组织映射模型的输入和输出层原则上都可以是任意的高维空间,通常输出都是一维或二维空间。学习过程中输出层各单元的位置并不改变,是各单元与输入层连接的权值在变化,映射后图中各单元的位置是用它们的权向量(与输入空间同一维数)表示的(称为该单元的虚位置)。同时,所谓的由高维空间映射到二维空间并不是映射为一个平面,而是映射到原输入高维空间中的一个超曲面;映射到一维也不是映射为一条直线,而是在原输入空间中的一个超曲线。

T. Kohonen 将这种自组织映射方法应用于语音识别,他构造了一种基于神经网络原理的声控打字机。按自组织映射方式工作的神经网络,实现芬兰语的音素分类,其语音识别率为 93% ~98%。现在自组织映射已在许多领域得到成功应用。

参考文献

1. Kohonen T. Self organizing maps. 3rd ed. New York: Springer-Verlag, 2001

2. Kohonen T, et al. Engineering application of SOM. Proc IEEE, 1996, 84: 1358-1383

3. 靳蕃,范俊波,谭永东. 神经网络与神经计

算机原理. 应用,成都: 西南交通大学出版社,1991

（阎平凡　靳蕃）

zifuji

字符集(character set)　按某种约定而设定的一组表示数据的符号。数据可以是数、英文字母、汉字、符号、命令、图形、图像、声音等。在字符集内,每个字符都有确定的二进制编码,并能被计算机识别。

在编码字符集中的字符代码必须具有: ①唯一性,即字符与二进制代码之间为一一对应关系;不存在 1 个字符有 1 种以上的代码,或 1 种代码表示 1 种以上的字符。②规范性,各种编码字符集都有相当大的适用范围,在此范围内的计算机用户必须严格遵守其规定。只有这样,才能保证信息的交换和相互利用。③兼容性,当一个计算机系统中采用 1 种以上的字符集时,应考虑不同字符集中字符代码的兼容性: 即一方面,要使各种字符共容于一个系统中,能够相互区分而不会混淆;另一方面,也要考虑不同字符集之间具有某种共性和继承性而不会相互冲突。

计算机中应用最广的编码字符集为美国制订的美国信息交换标准码(ASCII)。它采用 7 位二进制位进行编码。字符集中包含 32 个控制字符和 96 个图形字符,图形字符有数字、英文大小写字母和多种符号。与 ASCII 完全兼容的字符集有国际标准化组织制订的 ISO-646 及我国制订的 GB-1988。国外比较流行的另一种字符集为扩充的二-十进制交换码(EBCDIC),这是一种以 8 位二进制位编码的字符集,主要应用于美国 IBM 公司的大中型计算机系统中。为了适应汉字信息处理的需要,我国于 1980 年制订了“信息交换用汉字编码字符集 · 基本集”,其标准号为 GB 2312—80,简称国标码。它是我国应用最广的汉字编码字符集。GB 2312 包含汉字 6763 个,非汉字图形字符 682 个,每个字符以两个字节来编码。当在计算机中同时处理汉字及 ASCII 字符集的字符时,为了满足兼容性的要求,将 GB 2312 的代码作某种变化,即把字节的最高位设置为 1,以便同最高位为 0 的 ASCII 码相区分。为了提高字符集的覆盖能力,我国还制定了 GB 7589, GB 7590, GB 12345 等辅助汉字集,其中 GB 12345 主要用于繁体字。

除了 GB 2312 外,我国台湾地区还采用“通用汉字标准交换码(即 CNS-11643)”、BIG5 码及 TCA 码等。随着汉字在国际交流中的重要性日益提高,在国际标准化组织以及中、日、韩 3 国专家和政府的努力下,制定了包含中、日、韩 3 国所使用的汉字的编码字符集“统一的中日韩汉字辞汇与字序”,简称“CJK”大字符集。该字符集包含 20 902 个汉字,已成为“国际标准通用多八位编码字符集(ISO 10646)”的重要组成部分。我国所制定的 GB 13000—1993 字符集标准与之完全兼容。由于 ISO 10646 标准的完全实现比较困难。我国在 2000 年公布 GB 18030—2000 标准,该标准字符集中包括 27 484个汉字,完全覆盖并替代一度作为 GB 2312 标准重要补充的《汉字扩展内码规范》(即 GBK)。GB 18030—2000 标准作为国家强制性标准。

（黄震春）

zongxian biaozhun

总线标准(bus standard)　总线标准是国际公布或推荐的互连计算机系统中各个模块之间传送信息的公共通路的标准。它是把各种不同的模块组成计算机系统(或计算机应用系统)时必须遵守的规范。

总线标准为计算机系统(或计算机应用系统)中各个模块的互连提供一个标准界面,该界面对界面两侧的模块而言都是透明的,界面的任一方只需根据总线标准的要求来实现接口的功能,而不必考虑另一方的接口方式。按总线标准设计的接口是通用接口。采用总线标准可以为计算机接口的软硬件设计提供方便。对硬件设计而言,由于总线标准的引入,使各个模块接口芯片的设计相对独立,同时也给接口软件的模块化设计带来了方便。

每个总线标准都必须有详细和明确的规范说明,一般包括如下几个部分: ①机械结构规范,确定模板尺寸、总线插头、边沿连接器等的规范及位置; ②功能规范,确定各引脚信号的名称、定义、功能与逻辑关系,对相互作用的协议和定时进行说明; ③电气规范,规定信号工作时的高低电平、动态转换时间、负载能力以及最大额定值。

总线标准的制订通常有两种途径: ①某计算机公司(或生产厂)在发展自己的微机系统时所采用的一种总线,得到 OEM(原始设备制造厂)的普遍接受,按此总线规范开发相应的配套产品,进而形成一种为国际工业界广泛支持的实用总线标准; ②由专家小组在标准化组织的主持下从事开发和制订总线标准的工作,标准推出后即可由厂家和用户使用。

从事接纳和主持制订总线标准工作的有 IEEE

（美国电气与电子工程师协会），IEC（国际电工委员会）和 ANSI（美国国家标准局）组织的专门标准化委员会，这些委员会一方面为适应不同应用水平要求，从事开发和制定总线标准或建议草案；另一方面对现有的由一些公司提出的并为国际工业界广泛支持的实用总线标准进行筛选、研究、修改和评价，给以统一编号，作为对该总线标准的认可。

随着计算机系统的发展，总线在不断发展和完善，一些老的总线标准已不适应技术发展的需要，因而有的被淘汰，如 S-100 总线；有的被改进和完善，如 STD 总线。而新的总线标准也在不断产生，近 30 年来比较流行的总线有 IBM PC/XT 总线、ISA 总线、EISA 总线、VME 和 VME64 总线、MCA 总线、FutureBus 和 FutureBus + 总线、STD 和 STE 总线、VL-Bus、PCI 和 PCI-X 总线、PCI-E 总线、HT 总线和 InfiniBand 以及用于连接外设接口的 USB 总线（通用串行总线）、IEEE-1394 总线（FireWire，火线）、SCSI 总线（small Computer System Interface 小型计算机系统接口）等。其中，IBM PC/XT 总线是早期的计算机总线，是以 Intel 8088 微处理器为基础设计的，有 62 条信号线，其中 8 条数据线、20 条地址线。它采用大主板上带有扩展输入输出槽的结构。此外影响较大的有：

（1）ISA 总线　又称 AT 总线。它是在 IBM PC/XT 总线基础上增加 1 个 36 引脚的扩展槽，其数据总线宽度为 16 位，工作频率 8 MHz，数据传输速率最高为 16 MB/s。

（2）EISA 总线　ISA 总线的扩展。EISA 是 1988 年由 9 家计算机工业公司联合推出的。它与 ISA 总线完全兼容，并支持多个总线主控器，加强了 DMA 功能，增强了突发式传输，数据总线宽度增为 32 位，工作频率 8 MHz，最大的数据传输速率为 33 MB/s。最多可支持 6 个总线主控器，是一种支持多处理器的高性能 32 位总线标准。

（3）Futurebus 与 Futurebus +　是由 IEEE 开发的高性能总线标准，能在不同的多处理器系统中应用。其中 Futurebus + 是 Futurebus 标准的扩展，其差别的关键点有 3 个：①Futurebus 支持 32 位数据总线；Futurebus + 支持的数据总线宽度为 32 位、64 位、128 位和 256 位。②Futurebus 只支持分布仲裁协议，而 Futurebus + 支持分布和集中式两种模式。③Futurebus + 包含 3 位能力域，它允许模块声明它的能力，用以调节总线交换的重要模式。

（4）PCI 总线　PCI（peripheral component interconnect，外设部件互连标准）总线从结构上看，是在 CPU 和原来的系统总线之间插入的一级总线，由一个桥接电路实现对这一层的管理，并实现上下之间的接口以协调数据的传送。管理器提供了信号缓冲，使之能支持 10 种外设，并能在高时钟频率下保持高性能，为显卡、声卡、网卡和 MODEM 等设备提供连接接口，它的工作频率为 33 MHz/66 MHz。PCI 总线是一种不依附于某个具体处理器的局部总线，也支持总线主控技术，允许智能设备在需要时取得总线控制权，以加速数据传送。

（5）PCI-X 总线　2000 年，PCI-SIG 组织发表了新的、更快速的 PCI-X 总线，这是由 IBM、HP 和 COMPAQ 共同开发的，是 PCI 总线的一种扩展结构，在相同的频率下，PCI-X 将能提供比 PCI 高 14% ~ 35% 的性能。

（6）PCI-E 总线（PCI express 总线）　PCI-E 总线采用串行通信模式以及同 OSI 网络模型相类似的分层结构，该分层结构自上至下由内软件层、会话层、事务处理层、数据链路层和物理层组成，其具体的信号是两对低电压、分离驱动的电脉冲，一对负责传送，一对负责接收，并通过一个被称为 MSI（message signaled interrupt，基于通信信号的中断控制）的轮询方法来管理中断请求、电源管理请求和复位请求等系统信息。PCI-E 总线的主要优势就是数据传输速率高，目前最高可达到 10 GB/s 以上，而且还有相当大的发展潜力。

（7）InfiniBand 总线　是 NGIO 和 Futurebus I/O 技术的融合，采用全新的体系结构，与传统的 PCI 无法兼容。在 InfiniBand 系统中，远程存储器，网络和服务器之间的连接是通过一条位于中心的 InfiniBand 控制芯片和中继线完成的。采用 InfiniBand 通道设计，外部设备可以离服务器远达 10 km。根据不同需要，InfiniBand 标准为通道适配器设置 3 种工作方式，分别提供 500 MB/s，2 GB/s，6 GB/s 的带宽。InfiniBand 总线主要用于服务器系统中，使用 InfiniBand 总线的系统将会得到更高的带宽和扩展能力，增强系统的灵活性。

（8）HT 总线　HT（（hyper transport）总线采用点对点的单双工传输线路，引入抗干扰能力强的 LVDS 信号技术，命令信号、地址信号和数据信号共享一个数据路径，支持 DDR 双沿触发技术等等。HT 技术本质上是一种为主板上的集成电路互连而设计的端到端总线技术，目的是加快芯片间的数据传输速度。HT 总线被设计为两枚芯片间的连接，连

接对象可以是处理器与处理器、处理器与芯片组、芯片组的南北桥、路由器控制芯片等。

参考文献

1. 金兰,金波. 计算机组织:原理、分析与设计. 北京:清华大学出版社,2006

2. 孙德文. 微型计算机技术. 3 版. 北京:高等教育出版社,2010 (孙德文)

zonghe lilun xingneng

综合理论性能(composite theoretical performance, CTP) 美国政府为限制较高性能计算机出口所设置的运算部件综合性能估算方法。CTP 以百万次理论运算每秒 MTOPS 表示。CTP 自 1991 年 9 月 1 日起,启用根据估算结果及所定 CTP 数值限额,确定出口许可。停止使用原来设置的数据处理速率 PDR 值估算方法。

CTP 的估算方法为首先算出处理部件每一计算单元(如算术逻辑部件)的有效计算率 R,再按不同字长加以调整,得出该计算单元的理论性能 TP,所有组成该处理部件配置的计算单元 TP 的总和即为综合理论性能 CTP。

例如,定点加法单元的 $R=\dfrac{1}{3\cdot t_{定点+}}$,$t_{定点+}$ 为定点加法执行时间(μs);

定点乘法单元的 $R=\dfrac{1}{t_{定点\times}}$,$t_{定点\times}$ 为定点乘法执行时间(μs);

浮点 +, × 单元的 $R=\dfrac{1}{t_{浮+}},\dfrac{1}{t_{浮\times}}$ 等,$t_{浮+}$,$t_{浮\times}$ 分别为浮点加及乘的执行时间(μs)。

按操作数字长对 TP 加以调整:

$$TP = RL$$

$$L = (1/3 + WL/96)$$

式中,WL 为字长(b)。

对单个计算单元的处理部件:

$$CTP = TP$$

对由 n 个计算单元组成的多计算单元的处理部件:

$$CTP = TP_1 + c_2 \cdot TP_2 + \cdots + c_n \cdot TP_n$$

TP_1 为 n 个 TP 中的最高值。对共享存储的多计算单元的处理部件,其

$$c_2 = c_3 = \cdots = c_n = 0.75$$

c_i 为计算单元间互联强度系数。 (詹文岛)

zonghe yewu shuziwang

综合业务数字网(integrated services digital network, ISDN) 综合业务数字网是各种业务(例如电话、数据、图像等)都以数字信号进行传输和交换的通信网。它提供端到端的数字连接及标准的多用途的用户网络接口,实现通信业务的综合化。

由于早期存在的通信网络均按照业务的不同各自单独设立,如用于电话业务的电话网、用于电报业务的用户电报网、用于数据业务的数据通信网等。这种以不同的通信业务组网的方式都存在着需要独立的物理连接、终端,不同的用户网络接口、接入过程、寻址过程以及不同的运营管理部门等不足。无论对于用户还是管理部门,在投资、效率、方便使用以及运行管理和引入新的业务等方面存在问题。为了克服上述缺点,从根本上改变不同业务网络之间的分离状况,国际电信联盟 ITU-T 组织提出了综合业务数字网(ISDN)以解决只要通过一个网络标准接口,利用一条用户线就能实现各种通信业务。其中,64Kbps 的信息信道(称为 B 信道)和信号信道(称为 D 信道)分离。在一条 2B + D 的用户线上可以连接一部数字电话和一部数据通信终端同时工作。近年来 ISDN 已经完全被互联网取代。

参考文献

李正福. 综合业务数字网-ISDN. 北京:人民邮电出版社, 1993 (马妍 马严)

zubo luyou xieyi

组播路由协议(multicast routing protocol) 一种路由协议,规定在一个实际网络中实现组播数据包的转发时如何构造、维护组播路由(称为组播分布树),使组播数据包能够传送到相应的组播组成员。

根据组播路由位于自治域内还是位于自治域间而需要应用不同的组播路由协议。

主要的域内组播路由协议有 PIM-DM(密集模式协议无关组播路由协议)和 PIM-SM(稀疏模式协议无关组播路由协议)。运行 PIM-DM 协议的路由器周期性地发送 Hello 消息,发现邻接的 PIM 路由器,进行叶子网络、叶子路由器的判断,并且负责在多路访问网络中选举指定路由器 DR(designated router)。采用“扩散-剪枝”的方式进行组播数据包的转发。运行 PIM-SM 协议的路由器周期性地发送 Hello 消息以发现邻接的 PIM 路由器,在多路访问

网络中进行 DR 的选举。PIM-SM 通过显式地加入/剪枝机制来完成组播分布树的建立与维护，通过组播分布树进行组播数据包的转发。

域间组播路由协议是 MBGP（Multiprotocol Extensions for BGP-4）和 MSDP（Multicast Source Discovery Protocol，组播源发现协议）。

MBGP 能在整个互联网上组播路由策略，并能够在 BGP 自治域内或域间连接组播拓扑。MBGP 可以说是增强版的携带 IP 组播路由的 BGP，它携带了两组路由，一组提供单播路由，另一组提供组播路由。单播和组播路由信息可以通过同一个进程交换，但是存放在不同的路由表里。单播 BGP-4 协议所支持的常见的策略和配置方法都可以用到组播中。

在 MSDP 中，某个域内的汇聚点 RP（rendezvous point）与其他域内的 RP 建立 MSDP 对等关系，同这些对等关系交换信源信息。在 MSDP 里要求域内组播路由协议必须是 PIM-SM。

域间组播比较成熟的是 PIM-SM/MBGP/MSDP 组合方案，它实际上是 PIM-SM 协议在域间环境下的扩展。PIM-SM/MBGP/MSDP 增加了两个过程：①信源信息在 RP 集合中的泛洪，以实现信源和成员在 RP 点的会合；②域间组播路由信息的传递，目的是保证组播报文在域间的顺利转发。

IPv6 协议使用 PIM 作为域内组播路由协议，使用 MBGP 作为域间路由协议，没有相应的 MSDP 协议，而是是用嵌入的 RP。IPv4 和 IPv6 所使用的组播组管理协议的功能相似，但协议名称不同。IPv4 使用 IGMP（Internet Group Management Protocol）协议，而 IPv6 使用 MLD（Multicast Listener Discovery for IPv6）协议。

参考文献

1. 岩延，郭江涛. 组播路由协议设计及应用. 北京：人民邮电出版社，2002

2. 徐恪，徐明伟，陈文龙，马东超. 高级计算机网络. 北京：清华大学出版社，2012

（孙亚民）

zuhe luoji

组合逻辑（combinatory logic） 研究**组合子**的形式描述和性质的逻辑。大约从 1920 年开始，M. Schönfinkel 和 H. Curry 先后独立地发明了组合子，以后逐渐建立了关于组合子的形式系统，即**组合逻辑系统**。它原是作为高阶逻辑的子部分引入的，并试图作为无类型数学的一个基础。

组合逻辑系统的字母表由可数无穷多个变元以及常元 ***K*** 和 ***S***（称为组合子）组成。它的项归纳定义如下：(1)变元和常元是项；(2)若 *X* 和 *Y* 都是项，则(*XY*)是项；(3)每个项都可以通过有穷次应用(1)和(2)得到。不含变元的项称为组合子。如 ***SKK***，***S***(***KS***)***K***，***S***(***BBS***)(***KK***)和 ***SS***(***KI***)等都是组合子。项之间的弱化归关系 $\triangleright_w$ 和弱相等关系 $=_w$ 是组合逻辑系统中两个重要的基本概念。它们是分别由两组不同的公理和规则归纳定义的。组合逻辑系统的表达能力很强，可以在系统中定义自然数，进而表达递归函数，而且组合逻辑系统可定义的函数类与递归函数类相同，从而获得了计算的又一模型。

组合逻辑与程序设计语言密切相关。它其实就是一种抽象的程序设计语言，这可从 20 世纪 50 年代的 LISP 语言及后来的函数式语言看出。D. Scott 关于 **λ 演算**和组合逻辑的模型论，对计算机语言的形式语义学产生了重大影响。组合逻辑与 λ 演算关系紧密，它们几乎具有相同的能力

参考文献

1. Hindley J R, et al. Introduction to Cambinators and λ-Calculns. Cambridge University Press，1986

2. Curry H B, et al. Combinator Logic. Ⅰ. Ⅱ. North Holland，1985，1972

（宋方敏）

zuhe suanfa

组合算法（combinatorial algorithm） 计算对象是离散的、有限的数学结构的组合学问题的算法。组合算法的用途十分广泛。从方法学的角度，组合算法包括算法设计和算法分析两个方面。关于算法设计，已经总结出若干带有普遍意义的方法和技术，包括动态规划、回溯法、分支限界法、分治法、贪心法等。尽管如此，组合算法的设计仍然是一门艺术，需要高度的技巧和灵感。算法分析的任务是分析算法的优劣，主要是讨论算法的时间复杂性和空间复杂性。它的理论基础是组合分析，包括计数和枚举。计算复杂性理论，特别是 NP 完全性理论，与组合算法是紧密相关的。NP 完全性概念的提出，正是为了刻画包括旅行商问题、图着色问题、整数规划等在内的一大批组合问题的计算难度。计算复杂性理论研究算法在时间和空间限制下的能力以及问题的难度，使组合算法的研究有了更加清晰的框架，将组合算法的研究提高到一个新水平。

单纯形法 G. B. Dantzig 在 1947 年提出的一种线性规划算法，他本人以及其他学者后来又提出多

种形式的变形和改进。实践表明,单纯形法及其变形和改进是非常行之有效的,在市场上已经形成许多可以有效解决大型线性规划问题的软件包。线性规划研究线性目标函数在一组线性等式与线性不等式约束下的极值问题。这本来是连续问题,Dantzig发现线性规划问题的可行解集(即满足约束条件的点的全体)是一个超多面体。如果它的最优解存在,那么最优解一定可以在这个超多面体的某个顶点取到。由于超多面体的顶点只有有限个,从而使线性规划成为一个组合优化问题。单纯形法是按照一定的规划,从可行解集的一个顶点转移到另一个顶点,使得目标函数的值不断地得到改进,最后达到最优。尽管单纯形法一直使用得很好,但在最坏情况下它需要指数运行时间,从而使线性规划问题是否属于P类一度成为人们关心的问题。1979年前苏联数学家 Л. Г. Хачцян 提出一个多项式时间的线性规划算法——椭球算法,从而解决了这个问题。1984年印度数学家 N. Karmarkar 又提出一个新的更好的多项式时间算法——投影算法。

排序和检索 将给定的元素序列按照某种顺序关系重新排列成有序序列称作排序。例如将 n 个数组成的序列按照从小到大的顺序重新排列;将 n 个英语单词组成的序列按照字典顺序重新排列。在给定的集合中查找某个特定的元素称作检索。例如从给定的 n 个数中找到最大的数。排序和检索算法已经成为数据结构中不可缺少的部分,是计算机科学技术中最基本、使用最频繁的算法。正因为如此,它们也是研究得最细致的一类组合算法(参见**排序算法**)。

图与网络优化算法 组合算法中内容最丰富的部分。图论中的计算问题包括图的搜索、路径问题、连通性问题、可平面性检验、着色问题、网络优化等。图论中的著名算法有求最小生成树的 Kruskal 算法、求最短路的 Dijkstra 算法和 Floyd 算法、求二部图最大匹配(指派问题)的匈牙利算法、求一般图最大匹配的 Edmonds“花”算法、求网络最大流和最小割的标号法等。

贪心法与拟阵 贪心法是求解关于独立系统组合优化问题的一种简单算法,求最小生成树的 Kruskal 算法就是一种贪心法。但是,贪心法并不总能找到最优独立集。贪心法能求得最优独立集的充分必要条件是 L 为一个拟阵。事实上,求最大生成树是关于拟阵的组合优化问题,而二部图的所有匹配构成的独立系统 U 不是拟阵。

穷举搜索 组合算法要解决的问题只有有限种可能,在没有更好办法时总可以用穷举搜索的办法来解决,即逐个检查所有可能的情况。当情况较多时这样做是很费时的。实际上,并不需要机械地检查每一种情况,常常有可能提前判断出某些情况不可能取到最优解,从而可以提前舍弃这些情况。这样便“隐含地”检查了所有情况,既减少了搜索量,又保证不漏掉最优解。参见**回溯法**。

分支限界法 一种用于求解组合优化问题的排除非解的搜索方法。它的基本思想是:把问题分成若干个子问题,估计子问题的目标函数值的上界或下界。对于最大值问题,子问题的下界也是原问题的下界。当子问题的上界小于原问题的下界时,不可能在这个子问题中取得原问题的最优解,舍去这个子问题。否则将这个子问题再划分成若干更小的子问题,重复上述过程,直到没有需要检查的子问题为止。

其他组合算法还有动态规划、快速傅里叶变换等。

参考文献

1. Reingold E M, Nievegelt J, Deo N. Combinatorial Algorithms: Theory and Practice. Englewood Cliffs: Prentice Hall, 1977

2. Papadimitriou C H, Steiglitz K. 组合最优化算法和复杂性. 刘振宏,蔡茂诚,译. 北京:清华大学出版社,1988

3. 卢开澄. 组合数学:算法与分析. 上下册. 北京:清华大学出版社,1983

（张立昂）

zuhexue

组合学(combinatorics) 研究满足各种附加条件的有限个对象的集合(称为组态)的数学分支。又称组合数学、组合理论或组合分析。组合学研究的问题主要有:计数问题,即要求得到组态或组态等价类的个数;存在性问题,即希望判明某种特定组态是否存在,其中特别希望得到所论组态存在的简明充分必要条件;枚举、构造和算法问题,即具体列出要求的全部组态和给出得到这些组态的算法;优化问题,即根据某些明确的标准找出最优或近乎最优的组态。当然,为解决上述各类问题必须深入研究已知或未知组态的结构和内在性质。由于在数学、自然科学、管理科学的很多分支,尤其在计算机科学的理论和应用上经常要处理形形色色的组态,因此,它是近几十年来增长最快的数学学科之一。但就其

理论现状而言,组合学尚未显示出其统一性。目前在组合学中已形成了在对象和术语上自成体系的几个部分,最大同时也是与计算机科学联系最紧密的部分是**图论**,此外还有组合计数、组合设计、组合最优化和组合几何等部分。

组合学是一个古老而又年轻的数学分支。据传大禹治水时(约公元前2200年)曾见到"神龟"背上一种图案,用阿拉伯数字表示就是下列3阶幻方

$$\begin{matrix}4 & 9 & 2\\ 3 & 5 & 7\\ 8 & 1 & 6\end{matrix}$$

它的每行、每列以及二对角线上数字之和都是15。这一历史上最早出现的非平凡组态至少在东汉末年(公元200年前)已有确切记载。在11、12世纪,中国、波斯、印度的学者各自得到组合数(即n元集中r元子集的个数)的公式,其中宋代杨辉在《详解九章算法》(1261年)中载有11世纪人贾宪的"开方作法本源"图,即表示组合数间递推关系的三角形图表。17世纪以来,组合学逐渐成为一个独立的数学分支,这主要归功于欧洲的很多数学家在解决概率计算、正整数分析等数学问题和一些智力难题时的大量研究工作。其中德国数学家、哲学家L. W. 莱布尼兹在1666年出版了组合学的第一部著作《论组合的艺术》,他希望这一学科能应用到"所有科学领域",近代数学意义下的"组合"一词也源出于此。到20世纪30年代,组合学渐趋深入,一些专著相继问世,也得到了一批一般性的定理。20世纪60年代以来,以《组合理论杂志》在1966年刊行为标志,组合学进入蓬勃发展的新阶段。计算机技术的急剧发展和信息时代的到来,已经并将继续提出大量富有理论和实际意义的组合学课题,它们是推动组合学不断深入发展的巨大动力。

因为形形色色的组态无所不在,所以组合学的内容从来就很驳杂。这里简单介绍它的几个主要部分和两个有广泛影响的存在性定理。

图论　这是组合学中最大的一部分,有的文献把它当作与组合学并列的数学分支,其内容请参看**图论**条目。这里仅指出一点:大规模并行计算机系统的互联网络通常以有向或无向图为数学模型,称为网络拓扑,这时图的结点对应于处理机,图的边则对应于处理机间的通信链路,于是图论的研究与互联网络的设计和分析密切相关。

组合计数　其目标是确定一类组态中组态的个数。最基本的是排列与组合的计数公式:元素取自n元集S的不同的k个元的序列,个数是$(n)_k=n(n-1)\cdots(n-k+1)$,不一定不同的$k$个元的序列个数是$n^k$;$S$中$k$个不同元组成的子集个数是$\binom{n}{k}=(n)_k/k!$,$k$个不一定不同元组成的多重子集个数是$\binom{n+k-1}{k}$。另一经典计数问题是求一个$n$元集分拆成两两不交的$k$个非空子集之并的分拆方式数$S(n,k)$,称为第二类斯特林数,它有表示式$S(n,k)=\frac{1}{k!}\sum_{i=0}^{k}(-1)^i\binom{k}{i}(k-i)^n$。一般来说,很难求得用初等函数表示的计数公式,但已经发展了一些研究计数问题的方法,主要有:

(1) 利用生成函数　这是研究计数问题的主要途径,其基本思想是:为了获得有关一个有限或无限数列$\{a_k:\ k\geqslant 0\}=a_0,a_1,a_2,\cdots$的知识,用一个形式幂级数$g(x)=a_0+a_1x+a_2x^2+\cdots$来整体地表示这个数列,称$g(x)$为该数列的生成函数(也称母函数或发生函数)。再通过$g(x)$和对它的运算得到原数列的性质,而数列的一般项正是计数问题的解。如规定$\binom{n}{0}=1$,$k>n$时$\binom{n}{k}=0$,则数列$\left\{\binom{n}{k}:\ k\geqslant 0\right\}$的生成函数是$(1+x)^n$,由此$\binom{n}{k}$被称为二项式系数。令$x=1$即得恒等式$\sum_{k=0}^{n}\binom{n}{k}=2^n$,令$x=-1$得$\sum_{k=0}^{n}(-1)^k\binom{n}{k}=0$,还可以利用此生成函数推得不少关于二项式系数的恒等式。

(2) 利用递归关系　即对数列$a_0,a_1,a_2,\cdots$找到一种规则,使得从某一项起,数列的每一项可通过此规则由其前面的那些项唯一确定。例如,在各种问题中出现的斐波那契数列$\{f_k:\ k\geqslant 0\}$即可由递归关系$f_k=f_{k-1}+f_{k-2}(k\geqslant 2)$及初始值$f_0=f_1=1$完全确定。通过它可以求得此数列的生成函数$\sum_{k\geqslant 0}f_kx^k=4/(1-x-x^2)$,从而$f_k=\frac{1}{\sqrt{5}}(\alpha^{k+1}-\beta^{k+1})$,这里$\alpha=\frac{1}{2}(1+\sqrt{5})$,$\beta=\frac{1}{2}(1-\sqrt{5})$。

(3) 容斥原理和反演公式　已知有限集A以及A的n个子集$A_1,A_2,\cdots,A_n$,则A中不属于任一子集$A_i(i=1,2,\cdots,n)$的元素个数是$|A|-\sum_i|A_i|+\sum_{i<j}|A_i\cap A_j|-\sum_{i<j<k}|A_i\cap A_j\cap A_k|+\cdots+(-1)^n|A_1\cap$

$A_2\cap\cdots\cap A_n|$,这里$|S|$表示有限集 S 的元素个数。这就是最简单形式的容斥原理(也称筛式或逐步淘汰原理),用它可以求得一些计数公式。如错位排列,即 $1,2,\cdots,n$ 的共 $n!$个全排列中使得每个 $i(i=1,2,\cdots,n)$都不排在第 i 位的排列个数是 $n!-\binom{n}{1}(n-1)!+\binom{n}{2}(n-2)!-\cdots+(-1)^n\binom{n}{n}(n-n)!=n!\left(1-\frac{1}{1!}+\frac{1}{2!}-\cdots+\frac{(-1)^n}{n!}\right)$。上述容斥原理有诸多推广,其中最深刻的结果是 G. C. Rota 在 1964 年得到的局部有限半序集上的反演公式,它以容斥原理和数论上的默比乌斯反演公式为特例,以简明的方式统一了很多具体结论,是组合学的一个重大进展。

(4) 波利亚计数定理 有的问题中计数的对象不是组态,而是组态的等价类。例如,分别用红、蓝两种颜色来染一个可任意活动的立方体的 6 个面时,有理由认为恰有一面染成红色的 6 种染法是等价的,因为把立方体作一旋转就能把红面转成顶面。如果一种染法能通过立方体的适当旋转等同于另一种,则认为这两种染法是等价的。所以不等价的染法不是 2^6 种,而可以穷举所有情况得知共有 10 种。G. Pólya 在 1937 年得到的计数定理是迄今为止关于等价类计数的主要理论基础,他本人后来把这一定理的内容叫做"相对于给定置换群的不等价组态的计数"。对立方体染色的例子来说,如用 x,y 代表红、蓝色,则从波利亚定理可求得生成函数 $x^6+x^5y+2x^4y^2+2x^3y^3+2x^2y^4+xy^5+y^6$,它展示了把 6,5,4,3,2,1 和 0 个面染成红色的不等价染法分别有 1,1,2,2,2,1 和 1 种,总的不等价染法共 10 种。

(5) 渐近计数 绝大多数计数问题的解 a_n 的精确表示式无法求得或非常复杂,而在有的问题中人们主要关注当 n 趋于无穷时 a_n 的渐近性状。例如在算法分析中,人们更关注运算次数 a_n 当 n 趋于无穷时是按线性方式 cn,还是按 $cn^{\frac{3}{2}}$或 $cn^{\frac{1}{2}}\log n$ 方式增长?这时我们希望得到 a_n 的渐近式 $a_n\sim b_n$,其中 b_n 是比较简单的初等函数,且有 $\lim\limits_{n\to\infty}a_n/b_n=1$。著名的渐近式有斯特林公式 $n!\sim\sqrt{2\pi n}\left(\frac{n}{\mathrm{e}}\right)^n$,这里的 e 是自然对数的底。和前面几种方法相比,渐近计数主要依靠解析方法,尤其是复变函数论。

组合设计 主要研究特定组态的存在性问题和构作方法。20 世纪中叶以来,其内容日趋深广且渐成体系。下面是两种最有代表性的组合设计。

(1) 区组设计 最一般的情形可定义为 v 元集 X 以及 X 的 b 个子集 $B_1,\cdots,B_b$ 的族,其中各子集 B_i 称为区组,b 个区组不一定不同。很重要的一类特殊区组设计是平衡不完全区组设计,它进一步要求有正整数 k 和 λ,使得每个区组都是 k 元子集,且 X 的每一对不同元恰好同属于 λ 个区组。通常把这种设计称为(v,k,λ)设计,并假设 $1<k<v-1$。可证这时 X 的每一元恰好属于 $r=\lambda(v-1)/(k-1)$ 个区组,而区组总数 $b=\lambda v(v-1)/k(k-1)$。所以(v,k,λ)设计存在的必要条件是

$$\lambda(v-1)\equiv 0\pmod{(k-1)}$$

和

$$\lambda v(v-1)\equiv 0\pmod{k(k-1)}$$

但它们不是充分条件,例如(15,5,2)设计不存在。事实上,确定能使(v,k,λ)设计存在的 v,k,λ 的值的精确范围一直是研究这种组态的中心问题。当 $k=3,\lambda=1$ 时,T. P. Kirkman 在 1847 年和 M. Reiss 在 1859 年独立地证明了$(v,3,1)$设计(也称为斯坦纳三元系)存在的上述必要条件也是充分的。H. Hanani在 1961 年进一步证明了当 $k=3,4$ 时上述必要条件是充分的。对一般的 $k\geqslant3$,R. M. Wilson 则证明了上述必要条件的"渐近充分性",即对任意给定的正整数 $k\geqslant3$ 和 λ,存在常数 $v_0=v_0(k,\lambda)$,使得当 $v\geqslant v_0$ 且 v,k,λ 满足上述必要条件时,必定存在(v,k,λ)设计。

(2) 正交拉丁方 n 阶拉丁方可定义为数 $1,2,\cdots,n$ 排成的一个 $n\times n$ 矩阵,其中每一行和每一列上的 n 个数互不相同。两个 n 阶拉丁方 $\boldsymbol{A}=[a_{ij}]$ 和 $\boldsymbol{B}=[b_{ij}]$ 称为正交的,如果 n^2 个有序对(a_{ij},b_{ij}) $(i,j=1,\cdots,n)$互不相同。欧拉在 1779 年提出这个概念,并当 n 是奇数或 4 的倍数时构造出一对正交的 n 阶拉丁方,但他认为当 $n=4m+2$ 时不存在一对正交的 n 阶拉丁方。直到 1900 年 G. Tarry 才通过穷举所有6 阶拉丁方证明了欧拉的猜想当 $n=6$ 时为真。1958 年 R. C. Bose 和 S. S. Shrikhande,以及稍后的 E. T. Parker 分别找到了一对正交的 22 阶和 10 阶拉丁方,否定了欧拉猜想。接着三位数学家联手在 1959 年证明了一个令人惊异的结果:除 $n=2$ 和 $n=6$ 外,必存在一对正交的 n 阶拉丁方。另外在 30 年代已证明对任一正整数 n 至多有 $n-1$ 个两两正交的 n 阶拉丁方,且当 n 是素数幂时可具体给出 $n-1$ 个两两正交的 n 阶拉丁方。所以有待确定的最小阶是 $n=10$,但至今人们尚未能找到三

个两两正交的 10 阶拉丁方。

组合设计是在 20 世纪 20 年代数理统计中的试验设计和分析的推动下发展起来的，它的实际背景不满足于证明特定设计的存在性，还要求具体构造出这些组态。构造方法主要有借助于数论、代数、有限几何等数学工具以及计算机搜索的直接法和由小组态拼装成大组态的递归法。

两个存在性定理

（1）霍尔定理　设 $S_1, S_2, \cdots, S_m$ 是有限集 S 的 m 个不一定不同的子集的序列，如果 S 中有 m 个不同元 $x_1, x_2, \cdots, x_m$ 使 $x_i \in S_i (i=1, \cdots, m)$，则序列 $x_1, x_2, \cdots, x_m$ 称为子集序列 $S_1, S_2, \cdots, S_m$ 的一个相异代表序列，简记成 SDR。P. Hall 在 1935 年证明了下述有广泛影响的存在性定理：有限集 S 的子集序列 $S_1, S_2, \cdots, S_m$ 具有 SDR 的充分必要条件是，对任一正整数 $k \leqslant m$，子集序列中任意 k 个子集的并集至少含有 k 个元。

（2）拉姆齐定理　对任给的正整数 r, k 以及 $q_1, q_2, \cdots, q_k \geqslant r$，必存在数 N，使当 $n \geqslant N$ 时，对 n 元集 S 以及 S 的所有 r 元子集的集 $S^{(r)}$ 的任一有序分拆 $S^{(r)} = A_1 \cup A_2 \cup \cdots \cup A_k$，存在某个 $i, 1 \leqslant i \leqslant k$，及 S 的某个 q_i 元子集 S_i，使得 S_i 的所有 r 元子集都属于 A_i。这个有深刻内涵的组合学定理是F. P. Ramsey 在 1930 年发表的。当 $r=1$ 时它就是抽屉原理：把 S 的 n 个元任意分放进编号为 $1, 2, \cdots, k$ 的 k 个抽屉，则当 $n \geqslant q_1 + \cdots + q_k + k - 1$ 时，必在某个编号为 i 的抽屉中有至少 q_i 个元。当 $r>1$ 时它是一个典型的“存在性”定理。如记定理肯定其存在的数 N 的最小值为 $R(q_1, \cdots, q_k; r)$，称为拉姆齐数，虽经长期研究还只确定了极少几个拉姆齐数，例如至今尚未求得 $R(5,5;2)$ 的值。另一方面，在 1930 年前后，不同国家的数学家在不同的数学领域独立地发现了几个形色各异但却与拉姆齐定理有内在共性的定理，它们构成了现称为拉姆齐理论的组合学分支的基础。

组合最优化　根据明确的标准寻求最优或近乎最优组态的理论与算法，也称组合规划。其研究范围包括大量现实问题，如排序、工作安排、计划制定、装车、选址及路线设计等网络或图上的各种极值问题，这些问题在理论上还可以抽象概括为拟阵上的组合最优化问题。近二三十年来组合最优化是数学规划与计算复杂性理论的交叉领域。

组合几何　组合学和几何学的交叉学科，主要研究以特定类型的几何图形为对象的组态中的各种极值问题。此类问题的最早例子之一是：同时与一个实心球接触且与该球同样大小的实心球最多有几个？J. Kepler 和牛顿都认为是 12 个，但直到 20 世纪中期才严格证明了这一结论。“组合几何”这一术语大约在 20 世纪 50 年代正式出现，并逐渐发展成一个数学分支。

参考文献

1. 柯召，魏万迪. 组合论（上册）. 北京：科学出版社，1981

2. 魏万迪. 组合论（下册）. 北京：科学出版社，1987

3. Papadimitriou C H, Steiglitz K. 组合最优化：算法和复杂性. 刘振宏，蔡茂诚，译. 北京：清华大学出版社，1988

（李乔）

zuida gongyinzi

最大公因子（great common divisor）　能整除两个给定正整数的最大正整数。设 a, b 和 c 都是正整数，若 c 是 a 的因数，又是 b 的因数，则 c 称为 a 和 b 的公因子。如果对于 a, b 的任何公因子 d，都有 d 整除 c，则称 c 是 a 和 b 的最大公因子，记作 (a, b)。a 和 b 的公倍数中的最小者称为 a 和 b 的最小公倍数，记作 $[a, b]$，显然有 $ab = (a, b)[a, b]$。(a, b) 是 a, b 的所有公因子的倍数，$[a, b]$ 是 a, b 的所有公倍数的因子。若 $a = p_1^{\alpha_1} \cdots p_s^{\alpha_s}, b = p_1^{\beta_1} \cdots p_s^{\beta_s}$ 为 a, b 的因子分解式，$p_1, \cdots, p_s$ 为不同的素数，α_i 和 $\beta_i (i = 1, \cdots, s)$ 都为非负整数。记 $\gamma_i = \min(\alpha_i, \beta_i)$，$\delta_i = \max(\alpha_i, \beta_i)$，则 $(a, b) = p_1^{\gamma_1} \cdots p_s^{\gamma_s}$，$[a, b] = p_1^{\delta_1} \cdots p_s^{\delta_s}$。

给定两个正整数 a, b，一定存在唯一的整数 q_1 和 r_1，使 $a = q_1 b + r_1 (0 \leqslant r_1 < b)$，$q_1$ 称为以 b 除 a 的部分商，r_1 称为余数。继续进行这样的除法，设 $b = q_2 r_1 + r_2, r_1 = q_3 r_2 + r_3, \cdots (b > r_1 > r_2 > r_3 \cdots > 0)$，这一演算进行有限次后结束，最后得到 $r_k = q_{k+2} r_{k+1}$，r_{k+1} 就是 a, b 的最大公因子 (a, b)。上述计算最大公因子的方法称为欧几里得除法，也称辗转相除法。利用欧几里得除法，也可以找到两个整数 x, y，使 $ax + by = (a, b)$。

对于多项式，也可以类似地定义它们的最大公因子。设 $f(x)$ 和 $g(x)$ 为两个有理系数多项式，则必存在两个有理系数多项式 $q(x)$ 和 $r(x)$，使 $f(x) = q(x) g(x) + r(x)$，且 $r(x) = 0$，或者当 $r(x) \neq 0$ 时，$r(x)$ 的次数小于 $g(x)$ 的次数。同样可以利用欧几里得除法计算两个多项式的最大公因子，并将最大公因子表示成这两个多项式的线性组合。对于

任意域上的多项式这个方法也成立。 （裴定一）

zuida liu

最大流（maximum flow） 从源到目的地没有违反各边的容量约束且每个顶点的输入都等于输出的具有最大流值的流。为了严格定义最大流问题，首先必须定义流网络和流。所谓流网络 $G=(V,E)$ 是一个有向图，其中每条边 $(u,v)\in E$ 有一个非负容量 $c(u,v)\geqslant 0$。若 $(u,v)\notin E$，则 $c(u,v)=0$。并且它有两个特殊的顶点，即源 s 和目的地 t。为方便起见，假定对任意 $v\in V$，存在道路 $s\Rightarrow v\Rightarrow t$。故该图连通且 $|E|\geqslant|V|-1$。设 $G=(V,E)$ 是流网络，容量函数为 c，源和目的地分别为 s 和 t。G 中的流是实值函数 $f:V\times V\to\mathbf{R}$，它具有下面三条性质：①容量约束：对所有 $u,v\in V$ 有 $f(u,v)\leqslant c(u,v)$；②斜对称：对所有 $u,v\in V$ 有 $f(u,v)=-f(v,u)$；③流守恒：对所有 $u\in V-\{s,t\}$ 有 $\sum_{v\in V}f(u,v)=0$。这时，$f(u,v)$ 称为从顶点 u 到 v 的网络流。流 f 的流值定义为：$|f|=\sum_{v\in V}f(s,v)$。所谓最大流问题就是给定流网络 G，源 s 和目的地 t，希望求出从 s 到 t 的具有最大流值的流。更一般的，多源多目的地流网络的最大流问题可转换成上面的单源单目标流网络的最大流问题来求解。

由于管道中液体的流动、电网中电流的配送以及通信网络上信息的传输等现实问题均可抽象为求最大流问题，所以如何求解最大流问题一度成为研究的热点。先后提出了 Ford-Fulkerson 方法、Edmonds-Karp 算法、预流推进算法和提前算法。求解最大流问题的算法有许多应用。特别地，最大二分匹配问题可转换成最大流问题来求解。

参考文献

Cormen T H, Leiserson C E, Rivest R L. Introduction to algorithms. Cambridge, MA: The MIT Press, 1990

（殷建平 陈火旺）

zuiduan lujing wenti

最短路径问题（shortest path problem） 寻求加权图中两个指定顶点间加权长度最短的路径的问题（参见**图论**）。许多最优化问题都能划归为寻求某个加权图的最短路径问题，如运输网络的最低运费路线问题和最快运货路线问题，通信网络的最大可靠性信息传递路线问题和最大期望容量信息传递路线问题，设备更新问题，最佳库存效益问题，市场间运输路线优化表的制定问题，等等。因此对寻求加权图的最短路径，引起了人们的极大兴趣与关注。

给定加权图 $G=\langle V,E\rangle$，权函数 $w:E\to\mathbf{R}^+$（$\mathbf{R}^+$ 表示非负实数集合）及 $u,v\in V$。现在的问题是：G 中有无从 u 到 v 的路径，若有则求出一条加权长度最短者。迄今为止，标号法仍然是求解最短路径问题的最有效算法之一，其具体作法如下：

若 $i,j\in V$，则令

$$w_{ij}=\begin{cases}w(e), & \text{若 } e\in E \text{ 为从 } i \text{ 到 } j \text{ 的边}\\ +\infty, & \text{若无从 } i \text{ 到 } j \text{ 的边}\end{cases}$$

第 1 步 起点 u 标号为零且定标，其他顶点标号为无穷大且不定标。

第 2 步 对每个未定标顶点 j 给以临时标号 k_j 使

$$k_j=\min\{b_j,b_i+w_{ij}\mid i\in V\text{ 且 } i \text{ 已定标}\}$$

其中 b_j 为 j 的标号，b_i 为 i 的标号。

第 3 步 找出临时标号中的最小者，用它代替相应顶点（当有多个未定标顶点的临时标号均取最小值时，则任选其中之一）的原标号并定标，然后撤销所有临时标号。

重复进行第 2 步和第 3 步，直到终点 v 被定标或再没有未定标的顶点能被定标为止。用此法既可判断有无从 u 到 v 的路径，又能获得从 u 到 v 的加权长度最短的路径。且顶点 v 的标号（当 v 已被定标时）即为最短路径的加权长度。 （王兵山）

zuiruo qianzhi tiaojian fangfa

最弱前置条件方法（weakest precondition method） 基于最弱前置条件的一种程序完全正确性证明方法。**最弱前置条件**指保证一个语句执行正常结束并满足结果断言的最弱前提条件。它是一个谓词公式，通常用 $wp(\boldsymbol{S},\boldsymbol{R})$ 表示，这里，$\boldsymbol{R}$ 是语句 $\boldsymbol{S}$ 执行后所期望的结果断言（后置断言）。

E. W. Dijkstra 在前后断言的基础上提出了最弱前置条件的概念，以及相应的程序设计演算，使程序设计和程序验证可同时进行。

对于 E. W. Dijkstra 所定义的语言，语句的语义通过最弱前置断言给出。$wp(\boldsymbol{S},\boldsymbol{R})$ 可通过逆向推理导出。例如：赋值语句的语义是 $wp(x:=e,\boldsymbol{R})\equiv\boldsymbol{R}[x/e]$，即将 $\boldsymbol{R}$ 中 x 的所有自由出现同时代换成 $\boldsymbol{e}$。例如：

$$wp(\text{“}x:=x*x\text{”},x^4=10)\equiv((x*x)^4=10)\equiv(x^8=10)$$

为了证明循环的终止性，E. W. Dijkstra 引入了循环不变式和界函数。一般说来，一个循环呈如下形式：

{*invariant*: $\boldsymbol{P}$} —进入循环前，不变式 $\boldsymbol{P}$ 真，

{*bound*: $\boldsymbol{t}$} —并且 $\boldsymbol{B}$ 真时 $\boldsymbol{t}>\boldsymbol{0}$，t 是循环次数的上界

*do*B→Decrease $\boldsymbol{t}$，**S**true **od**

—当 $\boldsymbol{B}$ 真时，使 $\boldsymbol{t}$ 递减并执行 $\boldsymbol{S}$，$\boldsymbol{S}$ 执行过程真

保持 $\boldsymbol{P}$

{$\boldsymbol{P}\wedge\neg\boldsymbol{B}$} —则循环必然终止且终止时 $\boldsymbol{P}$ 真 $\boldsymbol{B}$ 假

若 $\boldsymbol{Q}$ 是 $\boldsymbol{S}$ 的执行能在有限时间内中止并满足 $\boldsymbol{R}$ 的任一前提条件，则必有 $\boldsymbol{Q}\Rightarrow wp(\boldsymbol{S},\boldsymbol{R})$。因此，证明前后断言 $\boldsymbol{Q}\{S\}\boldsymbol{R}$ 只需先求出最弱前置断言 $wp(\boldsymbol{S},\boldsymbol{R})$，再证明 $\boldsymbol{Q}\Rightarrow wp(\boldsymbol{S},\boldsymbol{R})$。

当给定了 $\boldsymbol{Q}$ 和 $\boldsymbol{R}$，根据 $\boldsymbol{Q}$，$\boldsymbol{R}$ 的结构，通过推导 $wp(\boldsymbol{S},\boldsymbol{R})$，可推出 $\boldsymbol{S}$ 的结构，从而将程序设计的过程变成数学推导的过程。例如，要设计一个循环 DO，使得当满足前置断言 $\boldsymbol{Q}$ 和结果断言 $\boldsymbol{R}$，则 $\boldsymbol{P}$，$\boldsymbol{t}$ 和 $\boldsymbol{B}$ 应满足 $\boldsymbol{Q}\Rightarrow\boldsymbol{P}\wedge bound:\boldsymbol{t}$，$\boldsymbol{t}\leqslant\boldsymbol{0}\Rightarrow\neg\boldsymbol{B}$ 及 $\boldsymbol{P}\wedge\neg\boldsymbol{B}\Rightarrow\boldsymbol{R}$。这实际上给出了循环语句设计的原则。

参考文献

1. Dijkstra E W. A discipline of programming. Englewood Cliffs, NJ: Printice Hall, 1976

2. Gries D. The science of programming. New York: Springer-Verlag, 1981

（伊波）

zuixiao erchengfa

最小二乘法（method of least squares） 用有限个线性无关函数的线性组合所做的最佳**平方逼近**。

根据给定的一组数据：(x_i,y_i)，$i=1,2,\cdots,m$，在某特定的函数类中找一函数 $f(x)$，使偏差平方和 $\sum_{i=1}^{m}[f(x_i)-y_i]^2$ 达到极小。有时，由于各点数据可靠性不一致，故可引进权数 $d_i>0$，$i=1,2,\cdots,m$，从而函数 $f(x)$ 可由求带权偏差平方和 $\sum_{i=1}^{m}d_i[f(x_i)-y_i]^2$ 达到极小而得出。这种找 $f(x)$ 的方法称为最小二乘法。它是 19 世纪初由 A. M. Legendre 和 C. F. Gauss 分别独立提出的。科学实验与工程技术中的许多经验公式大都应用此方法建立。一般取 $f(x)=\sum_{j=1}^{n}\alpha_j\varphi_j(x)$，其中 $\alpha_1,\alpha_2,\cdots,\alpha_n$ 为待定参数；$\varphi_1(x),\varphi_2(x),\cdots,\varphi_n(x)$ 为选定的一组线性无关的函数，称作基函数，且 $m\geqslant n$。令

$$c_{ij}=\varphi_j(x_i),\quad i=1,2,\cdots,m;j=1,2,\cdots,n$$

记

$$\boldsymbol{C}=\{c_{ij}\}\in R^{m\times n},\quad \boldsymbol{\alpha}=(\alpha_1,\alpha_2,\cdots,\alpha_n)^{\mathrm{T}},$$

$$\boldsymbol{y}=(y_1,y_2,\cdots,y_m)^{\mathrm{T}}$$

则最小二乘法可表示为：求向量 $\boldsymbol{\alpha}$，使

$$\|\boldsymbol{C\alpha}-\boldsymbol{y}\|_2^2 \tag{1}$$

达极小，由微分学极小值的方法可推得 $\boldsymbol{\alpha}$ 满足方程组

$$\boldsymbol{C}^{\mathrm{T}}\boldsymbol{C\alpha}=\boldsymbol{C}^{\mathrm{T}}\boldsymbol{y} \tag{2}$$

此方程称为最小二乘法方程组，它的解为最小二乘解。当 $\boldsymbol{C}$ 为列满秩时，方程组（2）有唯一解，若 $\boldsymbol{C}$ 不是列满秩，则（2）的解不唯一，但具有最小欧氏长度的解是唯一的。该解称为极小最小二乘解。不论何种情况，最小二乘问题（1）的解可表示为

$$\boldsymbol{\alpha}=\boldsymbol{C}^{+}\boldsymbol{y}$$

其中 $\boldsymbol{C}^{+}$ 为 $\boldsymbol{C}$ 的广义逆矩阵（在 Moore-Penrose 意义下）。但 m 较大时，$\boldsymbol{C}^{\mathrm{T}}\boldsymbol{C}$ 往往是病态矩阵，因而直接求解（2）是不利的，此时，可对 $\boldsymbol{C}$ 进行 $\boldsymbol{QR}$ 分解，即存在 $m\times n$ 阶正交矩阵 $\boldsymbol{Q}$ 及 $n\times n$ 阶上三角矩阵 $\boldsymbol{R}$，使

$$\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{C}=\begin{pmatrix}\boldsymbol{R}\\ \boldsymbol{O}\end{pmatrix}$$

则方程

$$\boldsymbol{R\alpha}=\boldsymbol{y}_1^{*}$$

的解就是所求的最小二乘解，其中 $\boldsymbol{y}_1^{*}$ 由向量 $\boldsymbol{y}^{*}=\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{y}$ 的前 n 个分量组成。

基函数 $\varphi_j(x)$ 的选取是极为重要的，最常用的方法如下。

多项式最小二乘法 取基函数 $\varphi_j(x)=x^{j-1}$，$j=1,2,\cdots,n$

$$f(x)=\sum_{j=1}^{n}\alpha_j x^{j-1}$$

但当 $n\geqslant 6$ 时的相应法，方程是病态的，计算时可取 $\varphi_j(x)$ 为正交多项式，或通过给定点列 $\{x_i\}_{i=1}^{m}$ 构造按点列正交的多项式。

样条最小二乘法 当数据 (x_1,y_1)，(x_2,y_2)，$\cdots$，(x_m,y_m) 较多且有较多波动起伏时，$f(x)$ 可假设为分段多项式，特别是三次样条 $S_3(t_0,t_1,\cdots,t_r)$，其中

$$\cdots<t_{-2}<t_{-1}<t_0<t_1<\cdots<t_r<t_{r+1}<\cdots$$

且使 $x_1,x_2,\cdots,x_m$ 都落在 $[t_0,t_r]$ 之上。令

$$\psi_{3,t}(x) = \frac{1}{2}\sum_{j=i}^{i+4}\beta_{i,j} | x - t_j |^3,$$

$$t = -3, -2, \cdots, r-1$$

$$\beta_{i,j} = \frac{4!}{\omega'_i(t_j)}, \quad \omega'_i(t_j) = \prod_{\substack{k=i \\ k\neq j}}^{i+4}(t_j - t_k)$$

取基函数

$$\varphi_j(x) = \psi_{3,j-4}(x), \quad j = 1,2,\cdots,n$$

则得到

$$f(x) = \sum_{j=1}^{n}\alpha_j\,\varphi_j(x) = \sum_{j=1}^{n}\alpha_j\,\psi_{3,j-4}(x),$$

$$t_0 \leqslant x \leqslant t_r = t_{n-3}$$

参考文献

1. Lawson C L, Hanson R J. Solving least squares problems. Englewood Cliffs, NJ: Printice-Hall, 1974

2. 冯康,等. 数值计算方法. 北京: 国防工业出版社,1978

(沈毅)

zuixiao shengcheng shu

最小生成树(minimum spanning tree) 由给定图的所有顶点和部分边形成的总权最小的树。具体来说,给定连通的无向图 $G=(V,E)$,其中 V 为顶点集,E 为边集,并给定权函数 W: $V\times V\to\mathbf{R}$,其中 $\mathbf{R}$ 为实数集,图 G 的最小生成树就是 E 的无圈子集 T 使得 T 连接所有顶点且其总权 $W(T)=\Sigma_{(u,v)\in T}W(u,v)$ 最小。显然,一个图的最小生成树可能不唯一。

由于输电线路的架设、公路体系的构建以及计算机网的连接等现实问题均可抽象为求最小生成树问题,所以如何形成一棵最小生成树的问题一度成为研究的热点。一种一般的求解思路是:将集合 A 初始化为空集∅,当 A 不是生成树时,求出对 A 安全的边 (u,v) 并把它加入 A 中,直到 A 是生成树为止。这里,边 (u,v) 对集合 A 安全是指 $A\cup\{(u,v)\}$ 仍是某棵最小生成树的子集。因此,求出对 A 安全的边 (u,v) 成为求解最小生成树问题的关键。可以证明,如果 $G=(V,E)$ 是连通的无向图,且有定义在 E 上的实值权函数 W,A 是 E 的子集并包含在 G 的某棵最小生成树中,$(S,V-S)$ 是重视 A 的 G 的任何割,(u,v) 是横过 $(S,V-S)$ 的轻边,那么边 (u,v) 对 A 是安全的。这里,无向图 $G=(V,E)$ 的"割"$(S,V-S)$ 是指 V 的一个划分。边 $(u,v)\in E$ 横过割 $(S,V-S)$ 是指它的一个端点在 S 中,而另一个端点在 $V-S$ 中。一个割重视边集 A 是指没有 A 中的边横过这个割。一条边是横过某个割的轻边是指其权是横过该割的所有边中的最小者。显然,轻边可能不唯一。根据上述结果求解时,每次挑选的安全边是轻边,所以形成的求解算法便是贪心算法。执行过程中,集合 A 总是无圈的。图 $G_A=(V,A)$ 形成一个森林。任意对 A 安全的边 (u,v) 连接 G_A 的不同连通分支。开始时 $A=\varnothing$,G_A 中有 $|V|$ 棵树,每次挑选一条边,森林中树的数目减 1,挑选出 $|V|-1$ 条边后,整个森林只含一棵树,这棵树就是要求的最小生成树,算法终止。从上述结果还可看出,定义不同的割,便可求出不同的轻边,从而得到不同的求解算法。历史上最有名的是克鲁斯卡尔算法和普列姆算法。在克鲁斯卡尔算法中,集合 A 是一个森林,加到 A 中的安全边总是连接两个不同分支的图的最小权边。而在普列姆算法中,集合 A 形成一棵树,加到 A 中的安全边总是连接边棵树与不在这棵树上的某个顶点的最小权边。

参考文献

Cormen T H, Leiserson C E, Rivest R L. Introduction to algorithms. Cambridge, MA: The MIT Press,1990

(殷建平 陈火旺)

zuiyouhua fangfa

最优化方法(optimization method) 寻求(逼近)某函数(目标函数)在某些附加条件(约束条件)下的最优(极大或极小)值的计算方法。最优化的两个主要分支是动态最优化(与时间有关)与静态最优化,后者通常称为数学规划。1939 年 П. Б. Канторович 首先研究和应用了线性规划方法,1947 年 G. B. Dantzig 提出了单纯形方法,他们为线性规划的最优化方法奠定了基础。1951 年 H. W. Kuhn 和 A. W. Tucker 给出了非线性规划的基本定理,为数学规划奠定了理论基础。随着计算机技术的飞速发展,最优化方法已在工程、军事、生产、管理和经济等方面得到了广泛的应用。运筹学中所处理的问题绝大部分是数学最优化问题。用来解决数学最优化问题的方法,例如数学规划(包括线性规划和非线性规划)、**排队论**与决策分析等都属于最优化方法范畴。最优化方法还涉及工程控制、最优控制、系统科学等。

无约束最优化方法 寻求(逼近)多元函数 $f(\boldsymbol{x})$ 在整个实 n 维空间 R^n 中的局部极小值点的计算方法称为***无约束最优化方法***。它也是许多约束最优化方法的基础。大多数无约束最优化方法都是迭

代法，每一次迭代都从某一点 $\boldsymbol{x}^{(k)}$ 移到另一个适合条件 $f(\boldsymbol{x}^{(k+1)}) < f(\boldsymbol{x}^{(k)})$ 的点 $\boldsymbol{x}^{(k+1)}$。为了得到 $\boldsymbol{x}^{(k+1)}$，首先要确定移动的方向 $\boldsymbol{S}^{(k)}$，其次要确定沿方向 $\boldsymbol{S}^{(k)}$ 移动的步长 λ_k，于是 $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \lambda_k \boldsymbol{S}^{(k)}$。对应于 $\boldsymbol{S}^{(k)}$ 与 λ_k 的不同选取，就得到不同的算法。算法可分为两大类：一类是直接法，一般对目标函数的解析性质不作苛刻要求，常用于变量不多、函数比较复杂或不易计算偏导数的情形。较为常用的直接法有鲍威尔法、单纯形法、模式搜索法和罗森布罗克法；另一类是解析法，要用到目标函数的(偏)导数。

最速下降法和牛顿法 在要求计算(偏)导函数值的算法类中，一般取移动方向 $\boldsymbol{S}^{(k)}$ 满足

$$(\boldsymbol{S}^{(k)})^{\mathrm{T}} \nabla f(\boldsymbol{x}^{(k)}) < 0$$

这里 T 表示矩阵(或向量)的转置运算，$\nabla f(\boldsymbol{x})$ 表示 $f(\boldsymbol{x})$ 在点 $\boldsymbol{x}$ 上的梯度，即

$$\nabla f(\boldsymbol{x}) = (\partial f(\boldsymbol{x})/\partial x_1, \partial f(\boldsymbol{x})/\partial x_2, \cdots, \partial f(\boldsymbol{x})/\partial x_n)^{\mathrm{T}}$$

最速下降法采取如下的迭代步骤：首先选取初始点 $\boldsymbol{x}^{(0)} \in R^n$ 及判别收敛的正数 ε，记 $\boldsymbol{S}^{(0)} = -\nabla f(\boldsymbol{x}^{(0)})$。其次，当 $\boldsymbol{x}^{(k)}$ 已知，且 $\boldsymbol{S}^{(k)} = -\nabla f(\boldsymbol{x}^{(k)})$，若 $\|\boldsymbol{S}^{(k)}\| < \varepsilon$，则取 $\boldsymbol{x}^{(k)}$ 为近似解；若 $\|\boldsymbol{S}^{(k)}\| \geqslant \varepsilon$，则采用单变量函数求极值的方法，并称之为线性搜索求 λ_k，使

$$\min_{\lambda \geqslant 0} f(\boldsymbol{x}^{(k)} + \lambda \boldsymbol{S}^{(k)}) = f(\boldsymbol{x}^{(k)} + \lambda_k \boldsymbol{S}^{(k)})$$

令 $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \lambda_k \boldsymbol{S}^{(k)}$，重复这一步骤，直到求得满足误差范围的近似最优解为止。在该方法中，取移动方向为 $\boldsymbol{S}^{(k)} = -[\nabla^2 f(\boldsymbol{x}^{(k)})]^{-1} \nabla f(\boldsymbol{x}^{(k)})$ 所构成的迭代方法称之为牛顿法。这里 $\nabla^2 f(\boldsymbol{x}^{(k)})$ 为 $f(\boldsymbol{x}^{(k)})$ 的二阶偏导数矩阵。

共轭梯度法和变尺度方法 该法既有较快的收敛速度又无须计算二阶偏导数。共轭梯度法是由 R. Fletcher 和 C. M. Reeves 于 1964 年提出的，首先从任意的初始点 $\boldsymbol{x}^{(1)}$ 开始，在第 k 次迭代时取移动方向

$$\boldsymbol{S}^{(k)} = \begin{cases} -\nabla f(\boldsymbol{x}^{(k)}), & k = 1 \\ -\nabla f(\boldsymbol{x}^{(k)}) + \dfrac{\nabla f(\boldsymbol{x}^{(k)})^{\mathrm{T}} \nabla f(\boldsymbol{x}^{(k)})}{\nabla f(\boldsymbol{x}^{(k-1)})^{\mathrm{T}} \nabla f(\boldsymbol{x}^{(k-1)})} \boldsymbol{S}^{(k-1)}, & k > 1 \end{cases}$$

然后再作线性搜索得到 λ_k 和 $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \lambda_k \boldsymbol{S}^{(k)}$。对于二次凸函数只要有限步就能达到最小值点 $\boldsymbol{x}^*$；对于非二次函数，常采用所谓周期性重开始的策略，这在理论上可以证明达到 n 步二阶收敛，若不采取这一策略，其收敛阶为线性的。变尺度方法首先由 W. C. Davidon 于 1959 年提出，后来相继形成两种常用的变尺度方法，分别称之为 DFP 算法和 BFGS 算法，它们的共同特点是：移动方向由 $\boldsymbol{S}^{(k)} = -H_k \nabla f(\boldsymbol{x}^{(k)})$ 确定，其中 H_k 为 n 阶对称正定方阵；H_1 可以任意选取，但通常取 $H_1 = I$。这两个算法只在 $H_k (k > 1)$ 的定义方法上有所不同。可以证明在一定的条件下，它们都是超线性收敛，且都是 n 步二阶收敛，后者与前者相比有较好的数值稳定性。

约束最优化方法 寻求(逼近)目标函数 $f(\boldsymbol{x})$ 在约束条件 $g_i(\boldsymbol{x}) \leqslant 0 (i = 1, 2, \cdots, m)$ 下的最优(极小)值点问题的计算方法称为**约束最优化方法**。这里 $f(\boldsymbol{x})$ 与 $g_i(\boldsymbol{x})$ 都是 R^n 中的实值函数，当 $f(\boldsymbol{x})$ 和 $g_i(\boldsymbol{x})$ 均为线性函数时，上述问题称为线性规划，否则称为非线性规划。非线性规划中的一个分支是凸规划，它的目标函数 $f(\boldsymbol{x})$ 与约束函数 $g_i(\boldsymbol{x})$ 都是凸函数。介于线性规划与凸规划之间的是二次规划，它的目标函数是正定二次型，约束是线性的。若对上述问题附加条件——某些变量或全部变量只能取整数值——则称为整数规划。此外还有随机规划、几何规划和多目标规划等。

求解线性规划问题的最常用算法是单纯形法。1963 年 P. Wolfe 将单纯形方法作了修改，用以求解二次规划，该法只经过有限次迭代即可求得最优解。1984 年 N. Karmarkar 提出一种关于线性规划的多项式时间的算法，这在理论上和应用上都是很有价值的。

可行方向法 该法根据逐次沿可行方向求可行解点的迭代思想构造一点列 $\{\boldsymbol{x}^{(k)}\}$，使其满足某种给定要求。其关键在于适当选取向量 $\boldsymbol{S}^{(k)}$，使点列 $\{\boldsymbol{x}^{(k)}\}$ 符合一般迭代法的要求。对线性约束的情形，当 $f(\boldsymbol{x})$ 为可微凸函数时有三种较为有效的求 $\boldsymbol{S}^{(k)}$ 的算法：①由 G. Zoutendijk 于 1960 年提出的可行方向法，取 $\boldsymbol{S}^{(k)} = \boldsymbol{y}^{(k)} - \boldsymbol{x}^{(k)}$，其中 $\boldsymbol{y}^{(k)}$ 为 $f(\boldsymbol{x})$ 在 $\boldsymbol{x}^{(k)}$ 处的线性近似函数 $f(\boldsymbol{x}^{(k)}) + \nabla f(\boldsymbol{x}^{(k)})(\boldsymbol{x} - \boldsymbol{x}^{(k)})$，在线性约束下达到最小值的最优解。②由 J. B. Rosen 于 1960 年提出的梯度投影法，运用在 $\boldsymbol{x}^{(k)}$ 处投影矩阵 $\boldsymbol{P}_k$ 的公式，取 $-\boldsymbol{P}_k \nabla f(\boldsymbol{x}^{(k)})$ 为 $\boldsymbol{S}^{(k)}$，从而避免了每迭代一次就要解一个线性规划的手续。③由 P. Wolfe 于 1963 年提出的既约梯度法，它应用消去基变量的方法和 $f(\boldsymbol{x})$ 的既约梯度的概念构造出 $\boldsymbol{S}^{(k)}$。这三种方法所产生的点列 $\{\boldsymbol{x}^{(k)}\}$ 虽然可以使函数值序列 $\{f(\boldsymbol{x}^{(k)})\}$ 单调下降，但却不一定收敛于最优解。随后陆续有不少研究工作，对原有方

法进行种种修正,从而得出具有收敛性的各种新方法。1969 年 D. Goldfarb 结合梯度投影法与变尺度法提出了一种可行方向法,对二次凸规划是有限步收敛的,而且可以推广处理非线性约束的情形。同年,由 J. Abadie 和 J. Carpentier 将既约梯度法加以推广,提出了广义既约梯度(GRG)法,用来求解具有非线性约束的最优化问题,它是解一般非线性规划问题的一种很好的算法。

上述线性近似型算法的收敛速度,一般都不高于超线性的。对于二阶可微的函数 $f(\boldsymbol{x})$,在 $\boldsymbol{x}^{(k)}$ 处若用二次函数 $f(\boldsymbol{x}^{(k)})+\nabla f(\boldsymbol{x}^{(k)})(\boldsymbol{x}-\boldsymbol{x}^{(k)})+\frac{1}{2}(\boldsymbol{x}-\boldsymbol{x}^{(k)})^{\mathrm{T}}\nabla^2 f(\boldsymbol{x}^{(k)})(\boldsymbol{x}-\boldsymbol{x}^{(k)})$ 来近似,进而对可微函数 $f(\boldsymbol{x})$ 用种种变尺度矩阵 $\boldsymbol{H}_k$ 去代替近似式中的二阶偏导数矩阵 $\nabla^2 f(\boldsymbol{x}^{(k)})$,将约束问题的求解化为求一系列二次规划的解,这类方法统称为序贯二次规划法。利用计算过程中得到的信息和变尺度公式来更新 $\boldsymbol{H}_k$,这种逐次二次规划算法也称为约束变尺度算法,它是求解带非线性约束的最优化问题的重要方法之一[1]。

序贯无约束极小化方法 该法将求解约束极值问题转化为一系列解无约束的极值问题。对于具有不等式约束 $g_i(\boldsymbol{x})\leqslant 0(i=1,2,\cdots,m)$ 的非线性规划问题,作函数 $p(\boldsymbol{x},t)=f(\boldsymbol{x})+t\sum_{i=1}^{m}[\max(g_i(\boldsymbol{x}),0)]^2$,在适当的假设下,$p(\boldsymbol{x},t)$ 是对 $\boldsymbol{x}$ 不加约束下的最优解 $\boldsymbol{x}(t)$,在 $t\to\infty$ 时趋于原问题的最优解。这种方法称为罚函数法。1969 年 M. R. Hestenes 和 M. J. D. Powell 结合拉格朗日乘子法和罚函数法的特点,对约束为等式的情形,提出了可微的增广拉格朗日函数,并指出在适当的假设下,能够求得原问题的最优解。此后陆续有不少工作对一般可微非线性规划构造了各种不同的可微增广拉格朗日函数,并给出了算法的迭代程序,这类方法统称为广义乘子法。鉴于罚函数法产生的点列 $\{\boldsymbol{x}(t_k)\}$ 是从约束集的外部来逼近约束边界上的最优解,提出所谓障碍函数 $B(\boldsymbol{x},r)=f(\boldsymbol{x})-r\sum_{i=1}^{m}\ln(-g_i(\boldsymbol{x}))(r>0)$,可使 $B(\boldsymbol{x},r)$ 的无约束最优解 $\boldsymbol{x}(r)$ 在约束集内达到,当 $r\to 0$ 时,$\boldsymbol{x}(r)$ 趋于原问题的最优解。这种方法称为障碍函数法。另外还有一种典型的障碍函数,其表达式为

$$B(\boldsymbol{x},r)=f(\boldsymbol{x})-r\sum_{i=1}^{m}(1/g_i(\boldsymbol{x}))$$

对兼有等式和不等式约束的最优化问题,可结合使用罚函数与障碍函数而构造出混合型函数来求解,1969 年 W. I. Zangwill 提出用统一的观点研究算法,他的基本思想是将算法视为一个点列集的映像。在一些假设下由上半连续的点到集映像产生的点列 $\{\boldsymbol{x}^{(k)}\}$ 收敛于最优解。从而统一了不少已有算法的收敛性的研究。这方面的工作还在不断发展。

参考文献

1. Fletcher R. Practical methods of optimization. Vol 1: unconstraied optimization. New York: John Wiley & Sons, 1979

2. Avriel M. 非线性规划——分析与方法. 李元熹,等译. 上海: 上海科学技术出版社,1979

(汪裕武)

zuoye

作业(job) 用户请求计算机系统完成的一个计算任务。它由用户程序、数据及其所需的控制命令组成,每个作业一般可分成若干顺序处理和加工的步骤,称为作业步。例如,某个用 FORTRAN 语言编写的计算问题,可经过以下作业步来求解:编辑、编译、连结、装入、运行。

在批处理环境下,一批作业有序地排列起来,分批输入计算机,再由作业调度选择并启动运行,形成了一个自动处理作业的*作业流*。用户根据系统提供的手段,对其作业在系统中的整个运行过程所进行的控制叫作业控制。作业控制由作业控制语言(JCL)来描述。JCL 由作业控制语句组成,每个 JCL 语句是一行传送作业步信息的编码,可穿在控制卡上,也可通过键盘以命令方式发送。

参考文献

孙钟秀,等. 操作系统教程. 4 版. 北京: 高等教育出版社,2008

(郑宇华)

zuoye guanli chengxu

作业管理程序(job manager) 操作系统中负责作业从提交到完成期间的组织、管理和调度工作的程序。

作业指用户提交给操作系统计算的一个独立任务。作业管理具有两类功能:作业调度和作业控制。按照一定调度策略,从输入井的后备作业队列中选择作业进入主存运行的工作称*作业调度*。它完成作业从收容状态到执行状态、从执行状态到完成状态

的转变。具体地说：按预定算法，从收容状态的后备作业中选择作业投入运行；记录进入系统的作业的情况，并为每个作业建立作业控制块；为选中作业分配所需资源，为该作业建立用户进程；做好作业运行结束后的善后处理。

选择和确定作业调度算法十分复杂，可用以下指标来评估其好坏：周转时间、响应时间、吞吐率和处理器利用率等。常用的作业调度算法有：先来先服务，最短作业执行时间优先，响应比最高优先，优先数法，分类调度法。

作业控制指用户使用操作系统提供的手段和设施来组织和控制用户作业的运行，即用户对作业上机操作的全过程的各种干预。例如，如何组织作业？如何输入作业？如何执行作业？如何处理异常？等等。可以分成两类作业控制方式：

（1）脱机作业控制　又称**批处理**。在这种方式下，用户应使用作业控制语言的语句，把要求计算机系统执行的工作写成作业说明书，连同程序、数据一起提交给计算机。当作业调度选择作业并投入运行后，作业控制程序逐条解释和执行作业控制语句，以实现对作业的控制。作业运行过程中，用户不得对作业进行干预，故称脱机控制。这种方式的优点是：处理器和资源利用率高、作业吞吐率大。批处理操作系统采用这种方式工作。

（2）联机作业控制　又叫交互型处理。在这种方式下，允许多个联机用户通过终端共享一台计算机系统，用户在联机终端上使用系统提供的终端操作命令，逐条打入，逐条执行，系统及时将执行情况和运行结果反映给用户，形成了一个人机组成的闭合系统，能充分发挥人的主观能动性，可动态、交互地控制作业运行。广泛应用于分时系统、个人计算机系统和工作站。

联机作业控制可分成以下三种：①键盘命令方式　用户通过键盘向计算机系统发出各种命令，系统执行用户的各种要求。②命令文件方式　用户按作业上机处理的控制步骤，把需键入的操作命令，事先编好输入并保存到一个文件中。这个文件称命令文件。其中，每一行都是一条键盘命令。执行命令文件时，只要按格式打入命令文件的文件名和附加参数，系统便可自动读取命令文件内容，并逐条解释执行，直至结束。③选单驱动方式　这种方式以层次式逐级选单为引导，用户不必键入命令，只需选择选单项便可执行各种交互式任务，给用户使用计算机系统带来极大方便，特别适合经验不足的程序员或非计算机专业人员使用。

随着多媒体技术的发展和成熟，趋向于利用声音、文字、图形、图像等形式进行数据 I/O 和联机作业控制。

参考文献

孙钟秀，等. 操作系统教程. 4 版. 北京：高等教育出版社，2008

（费翔林）

外 文 字 头

Ada yuyan

Ada 语言(Ada language) 一种通用**程序设计语言**。它具有如 PASCAL 等通用语言和某些专用语言的长处,有通用控制结构,有定义数据类型和分程序的能力,且易于控制并行任务和处理异常情况。所以它可用于数值分析计算,系统程序设计,并满足实时和并行操作等要求。Ada 语言既适于军用,也适于民用。

在 20 世纪 70 年代初,美国国防部为摆脱软件费用急剧增长的困境,提出设计研制统一的军用结构语言。其需求包括:可靠性、易维护性、结构化程序的构造、强数据类型、相对精度和绝对精度的规约、信息隐蔽和数据抽象、并发处理、异常处理、类属(式样)定义、与机器有关的设施等。美国军方认为没有任何现存的语言能满足所有这些需求,所以招标设计。1979 年 4 月最后选定由法国 Jean Ichbiah 教授领导的设计小组所设计的“绿色语言”,并取名为 Ada 语言,以纪念世界上第一位有文字记载的女程序员 Augusta Ada Lovelace(1815—1852 年)。

1980 年 7 月出版了 Ada 语言手册,年末提供了编译系统。1983 年 Ada 语言被正式列入美国标准(ANXI/MIL 1815A—1983),后来先后被批准为美国联邦标准和国际标准(ISO/IEC 8652:1987—1992)。美国国防部规定只有经测试合格者方可被命名 Ada 编译。从 1991 年起,许多国家的军方规定使用 Ada 语言作为唯一计算机程序设计语言。Ada 产品和应用不断增加。由于使用过程中的反馈意见,计算机硬件的飞速发展和软件技术的进步,如面向对象技术等,促使把 Ada 83 修改成 Ada 95,其内容要点如下:

(1) 扩展程序设计 允许继承、修改或增加父类型的已有分量和运算,目的是可以重复使用现有可靠的软件,不必重新编译和测试。

(2) 宽类程序设计 引入宽类类型以能处理同一类中的任何一种类型。至今我们所接触到的功能允许定义新类型为现有类型的扩展,例如我们引入了几种警戒类型,它们各不相同又相关联。现在引入宽类类型,就能操作任何一种警戒类型并相应地处理。

(3) 抽象类型和子程序 抽象类型的目的是为通过派生进一步建立有用的类型提供共同的基础。抽象子程序是一种占位符号,可以提供运算(它没有体)。

(4) 类型扩展 引入加标记类型的概念。以上都是有关类型扩展的,好处是不必重新编译、测试现有稳定的系统,可重复使用,这是面向对象语言最重要的特性。

(5) 动态选择 一种灵活的访问子程序机制,可以把子程序作为参数来传递,动态绑定(binding)。

(6) 其他访问类型 除了可访问子程序外,还可访问对象。

(7) 层次库 Ada 95 引入层次库,包含子代包和子程序。这样逻辑上不同的包可共享一个私有类型扩展包而不必重新编译。子代又可以有子代,带来许多灵活和方便。子代分公有、私有两种。

(8) 私有子代单元 私有子代的说明对用户命令是可见的,私有子代就不给用户任何附加的可见性,这便于把大程序分层。

(9) 保护类型 封装并提供对类型的永有对象数据同步访问,而不必引入附加的任务。保护类型为多个任务同步存取共享数据提供了有效的手段,这是实时系统的一个关键要求。

(10) 任务调度和计时 Ada 83 的调度规则对于汇合特别不能令人满意,Ada 95 允许更自由地优先度选择和调度规则。

(11) 类属参数 Ada 95 中能声明形式类属包。

(12) 其他改进 引入控制类型,给出初始化、终止以及用户定义的赋值,以及使用访问值判别项给出继承功能。一些关于数组的限制被放松。包含了对无符号整型的支持,提供了移位和逻辑运算。这些都大大地减轻了系统程序员的负担。

(13) 预定义库 由于 Ada 95 提供层次库的概念,所以有许多标准库中提供的附加预定义包已重新构选。

(14) 专门需要的附录 Ada 95 中有 6 个专门

需要的附录：①系统程序设计；②实时系统；③分布式系统；④信息系统；⑤数值；⑥安全性和保密性。

参考文献

1. Ada 95 Reference Manual. International Standard ISO /IEC - 8652:1995, Jan. 1995

2. Ada 95 Rationale. Intermetrics, Inc. Jan. 1995

3. 袁崇义，徐泽同，译. 程序设计语言 Ada 参考手册(Ada 83). 北京：科学出版社，1986

4. 中科院软件所 8 室 Ada 9×研究小组，译. Ada 9×项目报告：映象文件. 北京：计算机研究与发展，1993，(增刊)

（程虎）

ALGOL yuyan

ALGOL 语言(ALGOL language)　一族以 ALGOL 60 为主要代表的高级**程序设计语言**。出于交流算法程序，使之适合于在不同计算机上运行的需要，一批欧美学者在 20 世纪 50 年代共同设计并实现了国际代数语言 IAL(international algebraic language)，发布于 1958 年，后来改名 ALGOL 58，其中 ALGOL 是 ALGOrithmic language 的缩写。它的完善版本 ALGOL 60 于 1960 年诞生。来自欧洲的 Friedrich L Bauer, Peter Naur, Heinz Rutishauser, Klaus Samelson, Bernard Vauquois, Adriaan van Wijngaarden, Michael Woodger 和来自美国的 John W Backus, Julien Green, Charles Katz, John McCarthy, Alan J Perlis, 和 Joseph Henry Wegstein 于 1960 年 1 月 11 日至 16 日在巴黎开会，会上产生了"算法语言 ALGOL 60 报告"的正式文本。1962 年 4 月，部分 ALGOL 60 报告起草人在罗马聚会，旨在消除 ALGOL 60 原报告中遗留的问题，从而产生了 ALGOL 60 修改报告(The Revised Report of ALGOL 60)，署名的仍是原来的 13 人。该报告在同年 8 月份由 IFIP 第二技术委员会审核通过，随后又由 IFIP 领导机构批准，于 1963 年正式公布。值得注意的是，该报告列出了一批 ALGOL 60 中明知存在的语义不精确，但是未能解决的问题。由于各地在实现 ALGOL 60 时发现仍然有不少问题需要澄清，IFIP 第二技术委员会组织专家研讨多次，最终由 R M De Morgan, I D Hill 和 B A Wichmann 起草 ALGOL 60 改动报告(The Modified Report of ALGOL 60)，于 1975 年批准公布。这项工作后来并未产生多大影响。

ALGOL 60 的**语法**用一种上下文无关的巴克斯·诺尔形式体系 **BNF** 定义，而**语义**用自然语言描述。ALGOL 60 引进了对后来的高级程序设计语言产生深远影响的一系列构造。它是一种**命令式语言**，执行赋值语句会改变存储的数据。它又是一种**类型语言**，所有变量、常量和函数必须预先定义其类型，基本类型是整型、实型和布尔型。它引进数组概念以反映算法上常用的矩阵。在 ALGOL 58 的复合语句的基础上增添说明部分之后成为分程序，从而引进了作用域的概念。ALGOL 60 程序由复合语句和分程序的嵌套组成，程序运行进出于嵌套之间可造成分程序的进退，使作用域成为动态的，这不仅使变量的从属关系更直观，而且可以节省存储。ALGOL 的控制机制包括条件语句和循环语句，循环的条件可以在循环体中改变，从而能够应付各种复杂的情况。过程明确了按值和按名两种调用方式。它还允许递归。

许多后来出现的高级程序设计语言都部分或全部采用了 ALGOL 60 的重要构造，包括一些直到现在还在使用的语言。由于 ALGOL 60 对后世程序设计语言的影响巨大，产生了"ALGOL 式语言"(ALGOL-like language)的概念。然而，ALGOL 语言的缺点和不足也是明显的。由于它的语法是上下文无关的，而其程序的语义实际上是上下文有关的，因此，语义是用自然语言描述的，从而是非形式的，使得程序设计人员、程序阅读人员和语言实现人员对同一程序会有不同的理解。有些程序构造甚至是有歧义的。例如，嵌套条件语句 if a = b then if c = d then x: = 1 else x: = 2 既可以理解为 if a = b then (if c = d then x: = 1 else x: = 2)也可以理解为 if a = b then (if c = d then x: = 1) else x: = 2。

ALGOL 60 规定了各种程序成分"先说明，后使用"的原则。唯独标号为例外，它可以到处放，从而助长了 goto 语句影响程序的结构性，使得 Dijkstra 认为有必要撰写题为"goto 语句有害"的文章来纠正人们滥用这种语句。

ALGOL 60 的第一个实现由 Dijkstra 于 1960 年 8 月在 Elektrologica X1 计算机上实现。此后出现了一系列 ALGOL 60 实现系统。这些工作推动了编译技术的研究，特别是像栈和递归的实现等以后成为通用的技术。

ALGOL 60 的发布引起了国内的浓厚兴趣。在北京，由董韫美领导的一个小组 1960 年起对 ALGOL 60 报告进行了深入的学习研究，认为当时的 ALGOL60 文本存在诸多问题，因而自行设计一种能够避免这些问题的语言，即 **BCY** 语言，该语言于

1965 年底在 112 机上实现。

在南京,由徐家福领导的小组自 1960 年 9 月起学习以 ALGOL60 报告为主的若干文献,1964 年 9 月南京大学与华东计算所合作于 1965 年 6 月在 J-501 机上实现了国内第一个可用的 ALGOL 系统。

20 世纪 60 年代中期,国际信息处理联合会(IFIP)为确定 ALGOL 60 的后继公开征求新的语言方案。由荷兰计算机科学家 A van Wijngaarden 设计的 ALGOL 68 在竞争中获胜,并于 1968 年 12 月 20 日在慕尼黑举行的 IFIP 2.1 工作小组会议上正式通过,随后由 IFIP 领导机构批准公布。它的最后文本是"算法语言 ALGOL 68 修改报告",到 1975 年才发表。修改报告的首席设计人也是 A van Wijngaarden,其他作者为 B J Mailloux, J E L Peck, C H A Koster, M Sintzoff, C H Lindsey, L G L T Meertens 和 R G Fisker。

ALGOL 68 的语法用一个二级文法表示,称为 W 文法。其中第一级文法描述**元语言**,第二级文法描述 ALGOL 68 语言本身,但其中含有元语言中的元概念。因此,虽然这两级文法各自都是上下文无关的,它们的组合却可以生成无穷多条语法规则,从而具有很强的上下文有关描述能力。ALGOL 68 的**语义**用一种部分形式化的英语表示,写成抽象机的形式,属于操作语义。

ALGOL 68 采用正交设计,尽可能使每一种运算能作用于每一种数据类型的数据。它是一种强类型语言,它的类型称为模式。运算过程中允许有限的模式转换,称为强制,所有强制都体现在语法公式中。ALGOL 68 又是一种可扩充语言,通过模式说明、运算说明和(运算)优先说明可以定义新的模式和新的运算符,但没有数据封装和抽象数据类型。它允许运算符一名多用,但没有多类型机制。

ALGOL 68 的程序组织采用分程序嵌套结构。除了传统的条件、循环和 goto 语句外,它还引入了平行结构,并发控制采用 PV 操作,输入输出(统称传输)以文件操作为基础,允许用户自定义传输格式。

ALGOL 68 从一开始就引起异议。ALGOL 68 主设计师的学生-著名计算机科学家 E W Dijkstra 以"我爱我师,我尤爱真理"的精神与 Duncun, Garwick, Hoare, Seegmueller, Turski 和 Woodger 等人提出了他们的"少数报告",表示坚决反对。反对理由主要是认为该语言文本的晦涩难懂以及该语言无助于提高编程可靠性。后来,ALGOL 68 只在少数国家(主要是西欧)流行,并且未能成为国际标准化组织(ISO)的标准。而当年在与 ALGOL 68 的竞争中失败的 ALGOL W 却以 PASCAL 的名字获得了很大的发展。ALGOL 68 反对者的意见是有道理的。早期的程序设计语言追求功能强大、灵活,当复杂的编程带来越来越多的问题时,人们逐渐认识到,程序设计语言更重要的性质应该是简明、严谨和可靠。ALGOL 68 诞生之日正处于程序设计革命化变革的前夜,结构化程序设计和抽象数据类型即将主导计算机语言的设计,而 ALGOL 68 的设计者未能认识到这个浪潮。不过 ALGOL 68 含有许多很好的语言设计思想与语言构造,它们都是计算机科学的重要成果。

参考文献

1. Naur P. (ed.) The revised report on the algorithmic language ALGOL 60, IFIP, 1963; 算法语言 ALGOL 60 修改报告. 耿立大,译;许孔时,重译. 陆汝钤,周龙骧,重校. 计算机应用与应用数学,重庆

2. van Wijngaarden A. (ed) The revised report on the algorithmic language ALGOL 68. IFIP, 1975. 陆汝钤,译. 算法语言 ALGOL 68 修改报告. 北京:科学出版社,1982

3. 徐家福. ALGOL 漫谈. 南京大学学报(计算机科学专刊二),1979

4. 陆汝钤. ALGOL 68 导引. 电子计算机参考资料,1975(69、70):1-119

(陆汝钤　郑国梁)

A* suanfa

A* 算法(A* algorithm)　一种启发式搜索算法,对于存在解的状态搜索问题,能够较快地找到一条最佳求解路径,即最小费用解径。是一般有向图启发式搜索——A 算法的改进算法。

在**人工智能**领域进行问题求解时,通常可以把**数据库**的状态看作一个节点,把数据库所有可能状态的全体看作一个隐含的问题空间,并表示为一个有向图。图中每条弧代表对数据库的一个操作,它可以使一个状态转换为另一个状态。如果从代表初始状态的起始节点出发,有一条路径通向目标状态节点,则称此目标状态所代表的问题在给定的初始状态下有解,而从初始状态到目标状态通路上的每一条通路,即操作符序列,就构成了问题的一条解径,或解题过程。

对于存在有解的人工智能问题,虽然用盲目搜索方法,如宽度优先搜索法等,也能得到问题的最佳

解径,但是由于在搜索过程中需要展开过多的节点,因而限制了可求解问题的规模。另一方面,若一般性地利用与问题有关的启发信息进行搜索,虽可大大地提高搜索效率,但却不能保证一定能够找到最小费用解径。A^*算法就是要综合考虑路径本身的计算费用以及寻求路径所需的费用这两方面的因素,以适度增加搜索工作量为代价,来保证找到问题的一条最佳解径。

在启发式搜索中,需要有一种计算从各待扩展节点出发,达到某种预定指标的希望程度的方法,以便能优先扩展那些最有希望的节点。一种常用的方法就是定义称为估价函数的实值函数f。

A算法及A^*算法定义的估价函数$f(n)$如下:

设函数$f^*(n)$是从初始节点S出发约束通过给定节点n,然后到达目标节点t的实际最小耗费路径的费用,$g^*(n)$是从初始节点S到节点n的最小耗费路径的费用,$h^*(n)$是从节点n到目标节点t的最小耗费路径的费用,并且$f^*(n)=g^*(n)+h^*(n)$,即节点n在最佳路径上的实际费用$f^*(n)$等于从初始节点S到节点n的最佳路径的费用与从节点n到目标节点t的最佳路径的费用之和。那么$f^*(n)$的估价函数$f(n)$为

$$f(n)=g(n)+h(n)$$

其中,$g(n)$是迄今为止已找到的从初始节点S到节点n的最小耗费路径的费用,它被视为$g^*(n)$的一个估计,满足$g(n)\geqslant g^*(n)$;$h(n)$是$h^*(n)$的一个估计,其定义取决于与问题有关的某些信息,例如,从节点n到达目标节点t的距离;节点n与目标节点t的差异等。$h(n)$常被称为启发式函数。

在图搜索过程中,A及A^*算法利用节点的估价函数值$f(n)=g(n)+h(n)$对OPEN表中的待扩展节点进行排序。节点n的估价函数值$f(n)$越小,它处于最佳路径的可能性越大,越应排在序列的前面,优先扩展。但与A算法不同的是,A^*算法还进一步要求所定义的启发式函数$h(n)$为函数$h^*(n)$的下界,亦即对图中所有节点n,均有$h(n)\leqslant h^*(n)$。这样,如果给定问题确有解存在,那么A^*算法就可如此通过减弱启发式信息量,适当地降低搜索效率,确保找到问题的最佳解路径。例如,在极端情况下,即$h(n)=0$(肯定满足下界条件),$g(n)=d(n)$(节点n的深度)时,A^*算法等同于宽度优先算法,而宽度优先算法总能找到达到目标节点的最小长度路径。

下面,我们给出A^*算法的一般执行过程。

(1) 建立搜索图G和OPEN表(开始它们仅含初始节点S),并计算S节点的估价函数值$f(S)$;

(2) 建立CLOSED表(开始时为空表);

(3) 若OPEN表为空表,无解,失败退出;

(4) 取OPEN表中第一个元素为当前扩展节点n,并将其放入CLOSED表中;

(5) 若n为目标节点,有解,成功返回。其解为图G中从节点S到n沿指针所得的路径(指针由第(7)步确定);

(6) 扩展节点n,生成不是n的祖先的所有后继节点$\{m_i\}$,计算各后继节点的估价函数值$f(n,m_i)=g(n,m_i)+h(m_i)$,其中,$g(n,m_i)$是初始节点S通过n到m_i的费用,$f(n,m_i)$是S通过n,m_i到目标节点费用的估计。将它们作为节点n的后继节点添入图G中;

(7) 标记或修改后继节点集合$\{m_i\}=\{m_j\}\cup\{m_k\}\cup\{m_l\}$中各节点的指针。这里,$m_k$为已在OPEN表中出现过的待扩展节点,$m_l$为已在CLOSED表中出现过的已扩展节点,$m_j$为从未在二表中出现过的第一次生成的节点;

对于所有节点$m_j\in\{m_j\}$,建立m_j到节点n的指针,并将它们添入到OPEN表中;

对于所有节点$m_k\in\{m_k\}$,比较约束通过n的m_k节点的估价值$f(n,m_k)$与在扩展n之前已计算出的m_k的估价值$f(m_k)$(假设还有其他从S到n的通路),若$f(n,m_k)$较小时,则令$f(n,m_k)$为节点m_k的估价函数值,即$f(m_k)=f(n,m_k)$,并修改m_k的指针,使其指向n;

对于所有节点$m_l\in\{m_l\}$,操作同m_k节点。并且,当$f(n,m_l)$值较小时,还要把m_l节点放回到OPEN表中,以便能重新扩展m_l以及计算并修改其后继节点的估价函数值和指针;

(8) 将OPEN表中节点按f值从小到大顺序重新排列;

(9) 转向(3)。

为了提高A^*算法的执行效率,人们已经证明,如果启发式函数h能进一步满足单调限制条件,亦即,如果对于所有节点n_i和n_j(n_j是n_i的一个后继节点)总有

$$h(n_i)-h(n_j)\leqslant C(n_i,n_j)$$

并且

$$h(t)=0$$

式中,$C(n_i,n_j)$为n_i到n_j的弧费用;

t为目标节点。

那么，每当 A^* 算法选择某个节点进行扩展时，也就是它已找到了从初始节点 S 到达该节点的最佳路径。这样就没有必要反复地扩展 CLOSED 表中的节点，因而可以删除上面算法步骤(7)中的最后一步了。

参考文献

1. Nillson N J. 人工智能原理. 石纯一，等译. 北京：科学出版社，1984

2. 林尧瑞，马少平. 人工智能导论. 北京：清华大学出版社，1989

（杨莉　康建初）

BASIC yuyan

BASIC 语言(BASIC language)　一种简单易学，使用方便的**交互式语言**。BASIC 是 beginner's all-purpose symbolic instruction code(初学者通用符号指令代码)的缩略语。美国 J. G. Kemeng 和 T. E. Kurtz 于 20 世纪 60 年代初开始研制，1966 年正式推出。BASIC 最初是为初学计算机的人设计的，因而简单易学，小巧灵活，使用方便，既可作为批处理语言使用，又可作为分时语言使用；既可用解释程序直接解释执行，也可用编译程序编译成目标代码再执行。BASIC 具有交互会话功能，在程序执行过程中用户和机器可以相互问答，并可在程序执行暂停时插入新的语句执行。但 BASIC 不适用于编写较大的程序。

BASIC 程序由若干个相连的语句行组成，每一语句行的前面可以有一行号(在较早版本中每一语句行前均需有一行号)，每一语句行包含一个语句(在有些版本中允许包含多个语句)。各语句行(按其行号的顺序)依次执行。但可用控制语句来改变执行流程。语句可分为说明语句，输入输出语句，控制语句等几类，语句语法结构比较简单。数据类型只有简单类型与数组两种。在较早版本中变量名只能是一个字母或一个字母后接另一个字符，一个程序中最多允许使用 260 个变量名；数组只能用一个字母表示，从而最多只能使用 26 个数组。另外，用户还可以最多定义 26 个自定义函数。

下面是一个求 N！(N 的阶乘)的程序：

```
PROGRAM   FACT
  INPUT   N
    LET   F = 1
  FOR I = 1 TO N
      LET F = F * I
    NEXT   I
  PRINT   N;"的阶乘为";F
END
```

BASIC 问世后，人们对之作了多种扩充，在最初的扩充中，吸取了 COBOL 与 ALGOL 等语言的长处。目前的扩充涉及矩阵运算，图形处理，文件处理，字符串处理，以及结构化控制语句等。目前的 BASIC 比最初的 BASIC 无论在形式上，功能上，还是在规模上都有很大不同。

1978 年美国发布最小 BASIC 国家标准(标准号 ANSI X3.60—78)，1984 年该标准上升为国际标准(标准号 ISO 6373—84)，1987 年美国发布全 BASIC 国家标准(标准号 ANSI X3.113—87)。我国于 1991 年发布 BASIC 子集国家标准(标准号 GB 12856—91)。

参考文献

1. Kemeng J G，Kurtz T E. BASIC programming. 2nd ed. New York：John Wiley & Sons，1971

2. 中华人民共和国国家标准，GB 12856—91. 国家标准 BASIC 子集，1991

（徐宝文）

BCD ma

BCD 码(BCD code)　参见**数制**和**十进制算术运算**。　（王爱英）

BCY yuyan

BCY 语言(BCY language)　汉语拼音 bianyi chengxu chushi yuyan(编译程序初始语言)的缩略语。它是一个与算法语言 ALGOL 60 相类似的语言(参见 **ALGOL60 语言**)，于 20 世纪 60 年代初期由中国科学院计算技术研究所的一个小组所设计。

BCY 具有与 ALGOL 60 类似的基本语言成分，但是避免了当时已知存在于 ALGOL 60 中的漏洞。这些与计算工具无关的语言成分有：计算语句、转向语句、空语句、循环语句、条件语句、子程序语句、简变说明、场说明、开关说明和子程序说明等。

BCY 与 ALGOL 60 不同之处有，增加了为描述数字计算机上的计算过程用的其他语言成分，如：输入语句、印刷语句、鼓传送语句、带传送语句、停语句、求和语句、鼓说明、带说明和修改部分等。于是可以描述磁鼓、磁带、输入、输出设备的使用，以及描述在编译前对源程序所作的修改。

BCY 的其他特点包括：使用汉字定界符，如始、终、若、则、否则、转、对于、执行、步长、到、当等共 33

个。BCY 的表达式中还允许使用机器字,并可以对其中的字段进行运算。

BCY 语言的编译系统首先于 1965 年在中国科学院计算技术研究所的 119 计算机上实现。以后又分别在该所的 109 乙机、109 丙机、015 机,以及电子工业部华北计算技术研究所的 DJS-8 机、华东计算技术研究所的 655 机上实现。（董韫美）

Boltzmann ji

Boltzmann 机（Boltzmann machine, BM） 神经元间具有反馈作用的多层**人工神经网络**。它由输入层、输出层和隐层构成,网络没有明显的层次,神经元间相互连接,每一神经元可取 0 和 1 两种状态。在这种模型中,当神经元的输入加权和发生变化时,将引起神经元状态的变化和更新。这种更新在各个神经元之间是非同步的,可以通过概率统计方法来描述。由于网络的状态采用了统计物理学中的 Boltzmann(玻耳兹曼)概率分布,因此称为玻耳兹曼机。

考虑包含 n 个神经元的网络情形,第 i 个神经元的状态用 s_i 表示,它和第 j 个神经元之间的连接对称,即权值 $w_{ij}=w_{ji}$,没有自反馈。第 i 个神经元的状态 s_i 随机地被决定,其为 1 时的概率为 $p(s_i)$,满足统计决策规则:

$$p(s_i)=\frac{1}{1+\exp(-\Delta E_i/T)} \tag{1}$$

其中,T 是温度参数。假如上述统计决策规则反复运用于每一神经元,网络将达到热力学平衡,即状态保持不变,状态的分布服从玻耳兹曼分布。玻耳兹曼机可看作 **Hopfield 网络**的一种推广,隐层的出现说明了玻耳兹曼机与前向多层网络的相似性。

玻耳兹曼机是 1985 年由 G. E. Hinton 等人利用了统计物理学的概念和方法发展起来的神经网络模型,可看作是一种外界概率分布的模拟机。其基本思想是将物理学中随机模拟退火的方法用于神经网络的状态分析。玻耳兹曼机的信息处理基于概率分布和统计动力学,它可实现任意随机函数的逼近和解决约束优化问题。

模拟退火算法将组合优化问题与统计力学中的热平衡问题类比,开辟了求解组合优化问题的新途径。它通过模拟退火过程,可找到全局(或近似)最优解。模拟退火算法是基于 Monte Carlo 迭代求解法的一种启发式随机搜索算法。在对固体物质进行退火处理时,通常先将它加温熔化,使其中的粒子可自由运动,然后随着温度的逐渐下降,粒子也逐渐形成了低能态的晶格。若在凝结点附近的温度下降速率足够慢,则固体物质一定会形成最低能量的基态。对于组合优化问题来说,它也有这样类似的过程。组合优化问题解空间中的每一点都代表一个解,不同的解有着不同的代价函数值。所谓优化,就是在解空间中寻找代价函数(亦称目标函数)最小(或最大)的解。设 $S=\{s_1,\cdots,s_n\}$ 为所有可能的组合(或状态)所构成的集合,$C: S\to R$ 为非负目标函数,即 $C(s_i)\geqslant 0$ 反映取状态 s_i 为解的代价,则组合优化问题就是找最小的代价函数 C。模拟退火时把每种组合状态 s_i 看成某一物质系统的微观状态,而 $C(s_i)$ 看成该物质系统在状态 s_i 下的内能,并用控制参数 T 类比温度。让温度 T 从一个足够高的值慢慢下降,对每个 T,用 Metropolis 抽样法在计算机上模拟该体系在此参数 T 下的热平衡态,即对当前状态 S 做随机扰动产生一个新状态 S',计算增量 $\Delta C'=C(S')-C(S)$,并以概率 $\exp(-\Delta C/kT)$ 接受 S' 作为新的当前状态。当重复地随机扰动足够多的次数后,状态 S 出现为当前状态的概率将服从玻耳兹曼分布。由此看出,模拟退火算法基本上由三部分组成:①以一定的概率密度跃迁到新的状态,这个概率密度函数称为生成函数;②以一定的概率密度容忍代价函数的偶然上升,这个概率密度函数称为容忍函数;③以一定的冷却程式降低温度,这个等效温度是生成函数和容忍函数中的控制参量,确定所引入的随机扰动(噪声)的强度。

玻耳兹曼机除了可以解决约束组合问题外,还可以通过学习,模拟外界所给出的概率分布,实现联想记忆。玻耳兹曼机用作联想存储器时,需要用监督学习(参见**机器学习**),其主要步骤简略描述如下:①随机给定初始状态,包括选择初始温度及降温的规律,随机给定网络中连线权值的初值;②外加一个输入向量(把输入输出箝位于某一状态);③按模拟退火算法运行以达到平衡状态;④按选定的降温规律降温,再反复退火达到平衡,并计算此平衡状态下任意两个单元同时为 1 的概率为 p_{ij};⑤调整权值:仅对输入单元箝位,重复②,③步骤,并计算此平衡状态下任意两个单元 i 和 j 的状态同为 1 的概率 p'_{ij},则权值的修正量为 $\Delta w_{ij}=\eta(p_{ij}-p'_{ij})$,$\eta>0$ 为步长;⑥重复步骤②~步骤⑤,直到 w_{ij} 不再改变。在玻耳兹曼机的学习过程中,温度参数 T 是一个重要的参数。假如温度 T 低,网络只有很少几个可达到的状态,易陷入局部极小,学习非常困难;假

如温度 T 高,网络可达到的状态多,状态易变换,但学习算法收敛的稳定性差。为了快速使网络达到低温平衡,玻耳兹曼机的学习一般采用模拟退火过程,先设置高温,然后让温度 T 逐渐从高温到低温减小。

玻耳兹曼机作为一种随机动力学系统,具有较强的信息处理能力,是处理带有随机扰动信息的一种有效方法,已应用于**模式识别**、语音识别和优化问题求解。其主要的缺点是训练过程需要大量的计算,训练时间过长。为了加速玻耳兹曼机的学习过程,提出一些玻耳兹曼机模型的变形和学习算法的改进,改善玻耳兹曼机的行为。其中包括平均场理论和 H. Szu 提出的柯西(Cauchy)机等。玻耳兹曼机学习算法的改进,非对称结构玻耳兹曼机动力学行为的研究吸引着人们作进一步的探索。

参考文献

1. Ackley D H, Hinton G E, Sejnowski T A. Learning algorithm for Boltzmann machine. Cognitive Science, 1985, 9: 147-169

2. 史忠植. 神经网络. 北京: 高等教育出版社,2009

(史忠植 阎平凡)

C yuyan

C 语言(C language) 一种使用颇为广泛的程序设计语言。它由 AT&T 公司 Bell 实验室的 D. Ritchie于 1972 年至 1973 年间在类似于 BCPL(由英国剑桥大学的 M. Richards 于 1969 年设计并实现的系统程序设计语言)的 B 上设计而成,故命名为 C。首先在 DEC PDP-11 机上实现,著名的 **Unix 操作系统**就是用 C 书写的。

C 语言在很多方面继承和发扬了 20 世纪 60 年代出现的许多高级程序设计语言的成功经验和特色,它使用自由书写格式,具有丰富的数据类型,多种存储类别,一定程度的模块化结构,采用结构化的控制,函数参数传值,并支持分别编译等。主要特点是语言与运行支撑环境分离,语言规模小,相对简单,表示方法简洁,高度灵活,程序运行效率高,可移植性好;有不少操作直接对应于实际机器所执行的动作,在许多场合可代替汇编语言;大量使用指针,对运算时数据类型的一致性限制较少。

尽管最初是作为一种系统程序设计的工具语言设计的,但 C 已成功用于各个应用领域,是当前使用最广泛的一种通用程序设计语言。

1983 年美国国家标准学会(ANSI)的 X3J11 委员会开始进行 C 的标准化工作,国际标准化工作始于 1985 年(ISO/IEC JTC 1/SC 22 WG 14),1990 年公布的国际标准 ISO/IEC 9899 以美国国家标准 ANSI C 为基础,是第一个支持多八位字符集的程序设计语言国际标准。我国国家标准等同采用了 ISO/IEC 9899。为使 C 语言能更好地支持对全世界各国语言文字的处理,适应新的编码字符集国际标准 ISO 10647,以及进一步发展 C 语言,ISO/IEC JTC 1/SC 22 WG 14 还在继续工作,预计 1995 年可正式公布 C 国际标准的补篇。

参考文献

1. Kernighan B W, Ritchie D M. The C programming language. 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 1988

2. ISO/IEC 9899: 1990(E) International Standard for Information Systems — Programming Languages - C. ISO/IEC Copyright Office, 1990 SC 22/WG 14/N288; ISO/IEC 9899: 1990/Amendment 1: 1994(E)

(金益民)

CIP-L yuyan

CIP-L 语言(CIP-L language) 一种支持转换式程序设计的**广谱语言**。CIP-L 可描述从软件功能规约、设计规约、作用式程序到过程式或面向机器的高效程序等不同的抽象级;提供了从高抽象级向低抽象级转换的语言结构和语义规则。它是 70 年代中期由德国慕尼黑技术大学 F. L. Bauer 主持开发的"直觉引导的计算机辅助程序设计"项目的主要部分,用以支持转换式程序设计方法(参见**转换方法**),保证程序转换的正确性。

作为广谱语言,CIP-L 用代数类型描述功能规约,用计算结构刻画代数类型的模型,用模式描述算法和控制结构。代数类型也就是**抽象数据类型**,其公理用一阶逻辑公式表示。CIP-L 要求,代数类型中的任一对象均可经有限步基本运算生成。这种生成结构可用 CIP-L 的计算结构刻画。模式是一个算法架构,用以描述算法策略和控制策略,将一个模式中的代数类型名赋予具体的语义解释后即可得到具体的算法。模式语言又分成核心层和多个扩展层。核心层定义了表达式子语言,可按逻辑式或函数式风格书写程序,采用模型论方法定义其语义。扩展层包含了说明性子语言(参见**说明性语言**),命令式子语言(参见**命令式语言**),并行子语言和控制子语言,它们的语义刻画采用转换语义方法,即每一语法

结构的语义均可由转换规则在有限步内归结到核心层语义。

CIP-L针对转换式程序设计的需要而设计。语言覆盖了从功能规约到过程实现各个层次，功能很强，但同时造成结构繁杂，不易掌握。

参考文献

Bauer F L, et al. The Munich Project CIP: The wide spectrum language CIP-L. Berlin: Spring-Verlag, 1985

（伊波）

COBOL yuyan

COBOL语言（COBOL language） 一种用于事务处理的通用**程序设计语言**。COBOL是common business oriented language的缩略语。

1959年5月由美国政府部门、用户、制造商和其他部门参加的会议上决定成立CODASYL，并提出了用于事务数据处理的语言需求。要求面向用户，面向问题，与机器无关，要求用简单英语或类英语表示，并尽可能避免符号化。

1959年9月由CODASYL的一个委员会提出了COBOL初稿，其最终报告被采纳并于1960年4月公布，称为COBOL 60。

COBOL语言是最早出现的程序设计语言之一。最早应用于事务数据处理，并使用文件、记录、组项、初等项的数据描述方法。早期使用十分广泛。标准化活动开展最早且至今仍有活力。通过不断吸收新概念，采用兼容式的发展和"过时"元素的处理方法来不断更新语言版本，以适应事务数据处理领域的各种新需求。

COBOL语言将其核心和功能模块均分成若干级，故可以灵活地构作功能上具有积木式的但又是标准的适用于不同层次的语言实现系统，从而适应不同的需求。

COBOL程序由4个部组成。标识部用于刻画程序的标识性特征；环境部用于刻画程序和计算机环境有关的成分；数据部用于刻画数据（包括文件、记录、组项和初等项等）的定义的构成以及相关特征和属性；过程部用于刻画处理加工流程，采用节、段、语句的结构层次。整个程序具有类英语的描述特点，具有较好的易读性。

目前的COBOL国际标准版本是ISO COBOL 1989—1985。其第一版为推荐版ISO COBOL R-1989—1972，它由1个核心模块和7个功能处理模块（表处理、顺序存取、随机存取、分类、报表编制、程序分段和库）组成。其第二版为标准版ISO COBOL 1989—1978，它由1个核心模块和11个功能处理模块（表处理、顺序输入输出、相对输入输出、索引输入输出、分类合并、报表编制、程序分段、库、排错、程序间通信和通信）组成。其第三版即为ISO COBOL 1989—1985，它由7个必需模块（核心、顺序输入输出、相对输入输出、索引输入输出、程序间通信、分类合并、源正文管理）和4个任选模块（报表编制、通信、排错、程序分段）组成。

我国的COBOL国家标准采用相应的国际标准。

参考文献

1. 中华人民共和国国家标准　程序设计语言COBOL（GB 4092—83）北京：中国标准出版社，1985

2. ISO Standard 1989—1985（American National Standard COBOL X3.23-1985）

（钱树人）

C# yuyan

C#语言（C# language） 一种基于微软.NET框架的面向对象**程序设计语言**。它是由微软公司Anders Hejlsberg领导的小组于2000年设计并发布的，支持基于以公共语言基础结构（common language infrastructure，简称CLI）为核心的微软.NET框架的应用程序开发。C#可用于为主机和嵌入式系统编程，适合于分布环境下的组件开发。

C#的设计理念是简洁、现代、通用和支持面向对象。C#在语法上接近C/C++，但在设计理念上更接近Java。C#支持严格的类型检查（强类型）、数组越界检查、未初始化变量检测和自动无用存区收集等。与Java类似，C#不支持全局变量和全局函数，它们可通过静态类成员来实现；不支持多继承，多继承是通过接口来实现的。与Java不同的是，C#支持操作符重载机制；可在程序中显式指出不安全的地方采用指针。除了面向对象外，C#还支持命令式、类属（泛型）以及面向组件等多种编程风格。

从C#发布以来，其版本经历了1.0、1.2、2.0、3.0和4.0，每个版本都在前一版本基础上增加了一些新的功能。欧洲以及国际标准化组织分别发布了C#的规范文本：ECMA-334（2002、2006）和ISO/IEC 23270（2003、2006），目前，这两个标准规范仅针对C# 1.0、C# 1.2以及C# 2.0的功能给出了说明，针对C# 3.0和C# 4.0的规范还未制定。

参考文献

1. Karli Watson, Christian Nagel. C#入门经典. 5版. 齐立波,译;黄静校,审校. 北京:清华大学出版社,2010

2. Jon Skeet. 深入理解 C#. 2版. 周靖,朱永光,姚琪琳,译. 北京:人民邮电出版社,2012

(陈家骏)

C ++ yuyan

C ++ 语言(C ++ language) 一种面向对象语言。C ++ 语言最先由 AT&T 公司 Bell 实验室计算机科学研究中心的 B. Stroustrup 在 20 世纪 80 年代初设计并实现,它是以 **C 语言**为基础的支持数据抽象和面向对象风范的通用**程序设计语言**,至今仍在进一步演变发展。

C ++ 是对 **C 语言**的扩充,扩充的绝大部分来自著名语言中的最佳特性:从 SIMULA 67 中吸取了类,从 ALGOL 68 中吸取了算符一名多用、引用和在分程序中任何地方说明变量,综合了 Ada 的类属和 Clu 的模块特点,形成了抽象类,从 Ada、Clu 和 ML 吸取了异常处理,从 BCPL 中吸取了用“//”表示注释。

C ++ 保持了 C 的紧凑、灵活、高效和易移植性强的优点,它对数据抽象的支持主要在于**类**概念和机制,对面向对象风范的支持主要通过虚拟函数。由于 C ++ 既有数据抽象和面向对象能力,又比其他面向对象语言如 Smalltalk、Eiffel、Commonloop、CLOS 等的运行性能高得多,加上 C 语言的普及,而从 C 至 C ++ 的过渡较为平滑,以及 C ++ 与 C 的兼容程度可使数量巨大的 C 程序能方便地在 C ++ 环境中重用,使得 C ++ 在短短的几年内迅速流行,成为当前面向对象程序设计的主流语言。

C ++ 的标准化工作由美国首先发起。1989 年美国国家标准学会(ANSI)成立了 X3J16C ++ 标准化委员会,1991 年 6 月,以 X3J16 的国际小组为主成立了 C ++ 国际标准化工作组 ISO/IEC JTC1/SC 22/WG 21,两个组织决定联合工作,以一个标准文本同时作为 ANSI 和 ISO 的标准,C ++ 标准化工作划分为核心,扩充,库,环境,与 C 兼容性和国际化等专题小组,预计 1996 年可形成正式标准。

参考文献

1. Stroustrup B. The C ++ programming language. 2nd ed. Addison-Wesley,1991

2. X3J16,SC 22/WG 21:Working Paper for Draft Proposed International Standard for Information Systems— Programming Language C ++

(金益民)

Direct3D tuxing biaozhun

Direct3D 图形标准(Direct3D Graphics Standard) 微软公司基于通用对象模式 COM(Common Object Mode)的多媒体编程接口 DirectX(Direct eXtension)的一部分,运行在微软公司的 Windows 系列操作系统(Windows 95 或更新版本)和 Xbox 系列互动游戏平台上。Direct3D 通过绕过图形设备接口 GDI(Graphics Device Interface)直接执行底层硬件操作,为基于 Windows 的 PC 和 Xbox 上运行的交互应用提供高效能的图形编程界面。

Direct3D 是一套所谓“直接模式”(Intermediate Mode)图形编程界面,能够提供所有硬件支持的功能的接口,包括坐标系变换、裁剪、光照、材质、纹理、深度缓存等。Direct3D 定义了三种类型的抽象对象:设备、资源和交换链。设备包括硬件抽象层(Hardware Abstraction Layer, HAL)、软件参考层(Reference Layer)和插件式软件设备(Pluggable Software Device),大部分基于 Direct3D 开发的三维图形应用程序运行于硬件抽象层。定义在该层的设备既能充分利用图形硬件的加速功能,又隐藏了硬件相关的设备特性。每个设备可以包括一系列的资源如表面、二维纹理、盒纹理、体纹理、索引缓存和顶点缓存等。每个设备还必须包含至少一个交换链(swap chain),即一系列按顺序逐个提交到前台显示的虚拟帧缓存。

和 OpenGL 一样,Direct3D 定义了一个如何根据顶点、纹理、帧缓存和状态生成最后屏幕图像的过程,即所谓绘制流水线。Direct3D 同样也支持在绘制流水线中插入用户自定义的顶点着色器、几何着色器和像素着色器以实现各种复杂绘制效果。微软公司在 2009 年发布的 Direct3D 11 中加入了对可编程三角化(Tessellation)的支持,从而可以在图形卡上完成复杂的细分曲面(subdivision surfaces)和位移贴图(displacement mapping)计算,这使得人们可以以较低代价绘制具有丰富几何细节的物体。另一方面,Direct3D 11 中还加入了计算着色器以方便在图形卡上完成通用目的的计算。

Direct3D 源于一套由 RenderMorphics 公司开发的用于医学图像和 CAD 软件的图形 API。微软公司于 1995 年收购了 RenderMophics,并随 Windows 95 发布了第一个版本的 Direct3D。Direct3D 8.0 之前,Direct3D 在易用性上并不出色,功能上也难以和同时期的 OpenGL 相比,但从 Direct3D 8 开始,很多问题都得到了解决,Direct3D 也开始能够提供对可

编程着色器的支持。Direct3D 9 提供了对浮点纹理格式、多绘制目标以及在顶点着色器中使用纹理的支持。随 Windows Vista 发布的 Direct3D 10 增加了对几何着色器的支持。Direct3D 11 在 2009 年作为 Windows 7 的一部分发布,并提供了对可编程三角化、计算着色器、多线程绘制的支持。

参考文献

1. Frank D. Luna 著. DirectX 9.0 3D 游戏开发编程基础. 段菲,译. 北京:清华大学出版社,2007

2. Wikepedia. http://en.wikipedia.org/wiki/Microsoft_Direct3D

(任重)

DOS caozuo xitong

DOS 操作系统(DOS operating system)　用于微型计算机的一种磁盘操作系统。

DOS 是美国 Microsoft 软件公司与 IBM(国际商业机器公司)开发的,广泛运行于 IBM PC 及其兼容机上的磁盘操作系统,全名叫 MS-DOS。

20 世纪 80 年代初,IBM 公司开发 IBM PC,当其涉足微型计算机市场时,曾多方考察选择配合该机的操作系统,1980 年 11 月,IBM 和 Microsoft 正式签约,日后的 IBM PC 均使用DOS 为标准的操作系统。由于 IBM PC 大获成功,Microsoft 也跟着得到了飞速发展,MS-DOS 从此成为个人计算机操作系统的代名词,发展成个人计算机的标准平台。

在 IBM PC 机上所配的操作系统称 PC-DOS 或 IBM-DOS,是由 IBM 向 Microsoft 买下 MS-DOS 的版权,另外作了修改和扩充而产生的。

MS-DOS 最早的版本是 1981 年 8 月发表的 1.0 版,至 1993 年 6 月推出了 6.0 版。MS-DOS 是一个单用户微型计算机操作系统,4.0 版开始具有多任务处理能力。MS-DOS 的主要功能有:命令处理、文件管理和设备管理。命令处理对用户输入的键盘命令进行解释和处理;文件管理负责建立、删除和读写各类文件;设备管理完成各种外围设备,如键盘、显示器、打印机、磁盘和异步通信设备的输入输出操作。此外,MS-DOS 还具有系统管理和内存管理等功能。

MS-DOS 采用分层模块结构,按照功能划分,它由组成层次的 4 个模块组成,其结构如图 1 所示。

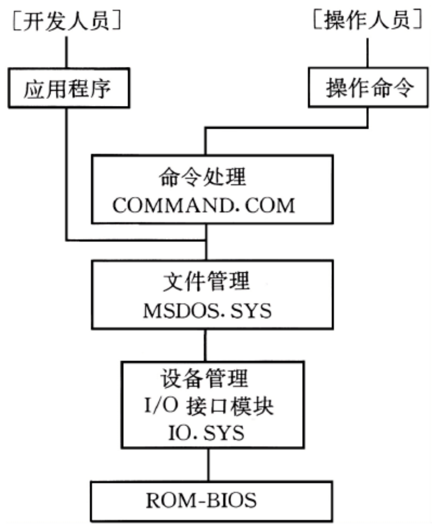


图 1　MS-DOS分层模块结构框图

(1) 基本输入输出系统　ROM-BIOS 存放在只读存储器中,它提供对 PC 机的 I/O 设备,如显示器、磁盘驱动器等最基本的 I/O 操作服务。ROM-BIOS 位于 DOS 的最底层,直接和硬件设备交互。它自身又由加电自测程序、I/O 支撑程序等组成。

(2) 输入输出接口模块　IO.SYS 是 MS-DOS.SYS 和 ROM-BIOS 的接口模块,它与 ROM-BIOS 共同处理 I/O 操作,统称为设备管理。其主要任务是:测定系统状态并进行系统初始化,管理和驱动各种外围设备,使磁盘系统复位,为引入内存的 MS DOS.SYS 重定位。

(3) 文件管理模块　MS DOS.SYS 是 MS-DOS 操作系统的核心部分,它提供系统与应用程序间的接口,其主要任务是:文件管理,存储器管理和提供例行程序服务。

(4) 命令处理模块　COMMAND.COM 位于 MS-DOS的最上层,直接和操作员打交道,接收从键盘来的输入命令,并确定如何处理这些命令。

MS-DOS中所有的信息都以文件形式存储在磁盘上,每个文件都有唯一的名字,以供识别。文件的名字由两部分组成:文件名和扩展名。前者由 1 至 8 个字符组成;后者是以圆点开始的,可取 1 至 3 个字符。文件名和扩展名可用的字符有:$,#,&,@,!,%,(、),{、},∧,~等。文件的扩展名用于对文件进行分类,补充说明文件的特性。MS-DOS中,有一批约定使用的扩展名,例如:.COM(命令文件),.EXE(执行文件),.BAT(批文件),.SYS(系统文件),.BAK(后备文件),.LIB(库文件)等。

文件目录是用来实现"按名存取"的主要手段,每个文件都有一个文件目录项,而文件目录项的集合就构成了文件目录。MS-DOS 的文件目录项由 32 个字节组成,如图 2 所示,其中包含了该文件的主要属性。

起始字节	字节个数	说　明
0	8	文件名
8	3	扩展名
11	1	文件可访问性
12	10	保留专用
22	2	建立或修改时间
24	2	建立或修改日期
26	2	起始簇号
28	4	文件字节数

图 2　MS-DOS的文件目录项

其中,文件可访问性区分是:读写文件、只读文件、隐含文件、系统文件或子目录信息等。起始簇号指明该文件的信息在磁盘空间中的第一个信息块的位置,1 个盘簇等于 2 个至 64 个扇区不等,每个扇区存放 512 个字节。建立文件时,MS-DOS自动在磁盘目录区建立这个文件的文件目录项。采用树型目录结构组织系统中所有文件的文件目录项,树的结点分 3 种:根结点、枝结点和叶结点。根结点表示根目录,枝结点表示子目录,叶结点表示文件目录项,亦即代表了这个文件。树中每个结点有一个名字以供访问,根目录是唯一的,其名字为反斜杠"\";子目录和文件的名字可由用户或系统给定,只要符合文件或子目录的命名规则即可。每个子目录均对应一个子目录项,其内容和文件目录项相似,包含了子目录的若干重要属性。

每个磁盘(软磁盘或硬磁盘)只有一个根目录,在磁盘初始化时由系统自动建立。根目录用来存放根目录下的所有文件和子目录的目录项,不同类型磁盘的根目录区大小不同,可从 4 个至 64 个扇区不等,因而,根目录中可容纳的目录项数就不同。由于根目录中可存放文件目录项,也可存放子目录项;子目录中可存放文件目录项,也可存放子目录项,而子目录中包含的目录项信息都存放在用户空间,这样,就形成了一个树型目录结构,只要有足够的存储空间,文件和子目录的个数就不受限制。

对于树型结构的文件目录,为了查找一个文件或子目录,必须给出这个文件或子目录的路径。路径是从指定目录或当前目录开始,到达指定文件或子目录所经过的路线。路径可以从根目录开始,也可以从当前目录开始,并由经过的子目录名组成,子目录名间用"\"分隔,如果路径末尾跟有文件名,也用"\"将文件名和它前面的目录名分开,这样形成了一条路径。

MS-DOS 按用户指出的路径来查找文件或子目录。以"\"开头的路径叫绝对路径,DOS 就从根目录开始查找,否则就表示相对路径,DOS 就从当前目录开始查找。

MS-DOS 的文件引用名可以看作是文件名的扩展,它是在文件名的前边加上盘符和路径,能进一步指明文件或子目录的具体位置,例如,A:　\XS DOS\WPS. EXE,B:　\FoxBASE\Fox. EXE,. . \LCH. COM,C:　\BC\BIN 等都是文件引用名的例子。

MS-DOS 的文件名字中可以使用通配符。通配符"?"代表文件名或扩展名中该符号所在位置的任一可用字符。通配符"*"代表文件名或扩展名中该符号所在位置的任一字符串。由于使用通配符的文件名字能代表一类文件,就可一次复制、删除或操纵一批文件,减少击键次数,提高操作效率,方便用户使用。

MS-DOS 文件的存取是由文件目录和**文件分配表**配合完成的。使用格式化命令格式化磁盘时,磁盘空间划分成 5 个部分:引导记录,FAT1,FAT2,根目录区和用户数据区。其中,文件分配表 FAT 是文件管理的重要数据结构,用以记录磁盘文件空间的使用情况。FAT2 是 FAT1 的复制品,以防 FAT1 被破坏而设置。FAT 的大小可变,例如,3. 5 英寸 1. 44 MB 盘,两个 FAT 每个占 9 个扇区;对大容量硬磁盘来说,每个 FAT 占 256 个扇区。FAT 的每个表目对应一个盘簇,当磁盘容量超过 10 MB 时,表目长 16 位。每个表目的状态指示它所对应的盘簇为:未分配、已分配、坏盘簇。FAT 记录了磁盘空间的使用情况及每个文件的数据占用磁盘空间的位置,由于每个表目对应一个簇,由簇号经过简单换算就可得到扇区号。每个文件占用的 FAT 表目采用连接方式链起来。因此,从该文件的文件目录项的起始簇号开始就可以最终实现文件信息的读写。

键盘命令是操作员从键盘上输入的,要求 MS-DOS 完成一定功能的会话命令。MS-DOS 的命令分成两类:

(1) 内部命令　直接嵌入 MS-DOS 的系统驻留区中,执行时不占用内存用户区,一经接收,就能立即执行。

(2) 外部命令　存放在磁盘上的各种命令文件,每当执行时,从磁盘上调到内存用户区,MS-DOS 把具有扩展名为 . COM,. EXE 和 . BAT 的文件都视作外部命令。

MS-DOS 命令可分成以下几类:基本DOS 命令、

文件操作命令、目录管理命令、批处理命令、高级DOS命令等。

汉字磁盘操作系统（CC-DOS）是为了使原来只具有西文处理能力的PC机DOS系统同时又具有处理中文信息的能力，在1983年，由我国电子工业部第六研究所，在MS-DOS的基础上，专为PC机开发成功的汉字操作系统。CC-DOS的开发和使用，为在我国更广泛地普及和使用微型计算机打下了坚实基础，具有很重要意义。

MS-DOS和CC-DOS的根本区别在于是否支持汉字字符处理，CC-DOS在MS-DOS的基础上工作，除保留原来MS-DOS的几乎全部功能以外，又增加了：汉字输入、汉字存储和处理、汉字输出功能。

汉字输入是汉字处理技术中最关键又最困难的部分，CC-DOS提供了面向用户的多种输入方法。面向一般用户的简单输入法有：拼音码、首尾码、五笔字型输入等；面向专业操作员的快速输入法有：国标码、区位码、电报明码等。

汉字字符在计算机中以变形国标码来存储，需用双字节，为了与西文ASCII码相区别，其中每个字节的最高位均为1。CC-DOS把汉字当作西文字符一样进行处理，汉字可使用到文件的各级，汉字可作为文件名和命令名。各种程序设计语言和应用程序中，汉字可作为变量名、字符串并和西文字符混杂处理。

汉字输出采用字形码，CC-DOS包含有一个汉字字库CCLIb，存放不压缩字形码，16×16点阵汉字字形码使用32个字节保存一个汉字的字形，含有GB2312—80规定的6763个汉字和682个图形符号。16×16点阵字形主要供屏幕显示，每个汉字的高度和宽度比ASCII字符大一倍。为了改善字形，又提供24×24点阵或48×48点阵字形码，由于点子密，字形更为美观，但点阵占用存储空间多。

除了汉字库CCLIB外，CC-DOS还有两个核心文件，以CC-DOS 4.0为例，分别称FILE1.EXE和CCCC.EXE。FILE1.EXE的功能是做好装入汉字库CCLIB的准备工作，检查汉字库的完好性，为汉字库申请内存区域，初始处理及进行模式转换。CCCC.EXE的功能是将汉字库装入申请好的内存区域，完成把ROM-BIOS改造成CC-BIOS，使其能支持各类I/O设备的汉字输入输出，再配上原有的MS-DOS操作系统，就可以使用键盘、显示器进行汉字的输入输出了。

在CC-DOS控制下工作时，有三种操作方式：纯西文方式、中西文方式和纯中文方式。可以通过键盘控制键控制操作方式的切换。

MS-DOS在20世纪90年代还在继续发展，1991年推出DOS 5.0版，1993年6月推出DOS 6.0版。DOS 5.0增加了改进的图形用户界面DOS SHELL；扩大了内存管理能力，可利用640 KB以上高地址区存放部分系统驻留程序；提供多任务处理能力，使PC能运行前后台任务；直接支持高达2 GB的硬磁盘分区；CPU扩充了保护模式，通过扩充和扩展内存驱动程序，使PC 286可访问16 MB内存，PC 386以上可访问4 GB内存；增加了DOSKEY，SETVER，HELP等10多条新命令；支持3.5英寸2.88 MB软磁盘。DOS 6.0具有DOUBLESPACE压缩磁盘功能；配置CD-ROM驱动程序；帮助设施改为全屏幕交互工作方式；具有CONFIG.SYS多重配置功能；进一步改进和增强内存管理功能；增加了消除磁盘碎片及删除整个子目录命令；增加INTERLNK，使两台微型计算机能直接相连并共享磁盘和打印机；MSAV工具用来检测和消除病毒；MSD工具能报告机器硬件及内存工作状态；POWER工具用来降低程序和机器处于等待态时的功耗。

参考文献

1. Microsoft MS-DOS Operating Version 4.0, Microsoft Corporation, Docoment Number 410630001-400-R10-1088. 1988

2. Norton P. Inside the PC. Brady, A Division of Prentice Hall Computer Publishing Inc., 1993

（费翔林）

DPLL fangfa

DPLL方法（Davis-Putnam-Logemann-Loveland algorithm）　一种完备的判定命题逻辑中合取范式（CNF）可满足性的算法。在求解可满足性（SAT）问题上，DPLL方法较**归结方法**更容易获得可满足性的模型，而归结方法则更容易给出不可满足性的证明。

1960年Davis和Putnam共同发布了DP过程，其中包含了单文字规则，纯文字规则和消去规则。由于消去规则容易导致子句长度的快速增加，并且不容易产生单文字子句，所以1962年，Davis、Logemann和Loveland发布了一个新算法，用分裂规则取代了消去规则，这个算法通常称作DPLL方法。DPLL方法是一个相当高效的过程，即使经过了近50年的发展历程，它目前仍是大多数有效而完备的SAT求解器以及部分一阶逻辑定理证明器的核心部

分。其中比较著名的有 GRASP, SATO, Chaff, MiniSAT 等。

DPLL 方法包括以下三个规则：

单文字规则：如果子句集 Φ 中有一个单文字子句 L，删除 Φ 中包含 L 的所有子句得 Φ'，则①若 Φ' 为空，则 Φ 可满足。②若 Φ' 非空，在 Φ' 中删除所有文字 $\neg L$ 得 Φ''，则 Φ 不可满足当且仅当 Φ'' 不可满足（若在 Φ' 中有单文字子句 $\neg L$，则删除 $\neg L$ 得空子句□）。

纯文字规则：说子句集 Φ 中的文字 L 是纯的，当且仅当文字 $\neg L$ 不出现在 Φ 中。如果子句集 Φ 中的文字 L 是纯的，删掉 Φ 中所有包含 L 的子句得 Φ'，则①若 Φ' 为空，则 Φ 可满足。②若 Φ' 非空，则 Φ 不可满足当且仅当 Φ' 不可满足。

分裂规则：如果子句集 Φ 可写成如下形式：$(A_1 \vee L) \wedge \cdots \wedge (A_m \vee L) \wedge (B_1 \vee \neg L) \wedge \cdots \wedge (B_n \vee \neg L) \wedge R$，其中 $A_i(i=1,\cdots,m)$，$B_j(j=1,\cdots,n)$，R 中都不含 L 或 $\neg L$。令 $\Phi_1 = A_1 \wedge \cdots \wedge A_m \wedge R$，$\Phi_2 = B_1 \wedge \cdots \wedge B_n \wedge R$，则 Φ 不可满足当且仅当 Φ_1 和 Φ_2 同时不可满足。L 称为分支文字。

对于一个给定的子句集，DPLL 方法重复地对这个子句集使用这三条规则，直到得出子句集可满足或不可满足的结论。

应用分裂规则时，分支文字的选择对算法效率有很大的影响。现在对 DPLL 方法的进一步改进主要体现在如下几个方面：定义不同的分支策略、定义新的数据结构、定义回溯算法的变体等。其中回溯算法的变体又包括非顺序回溯和子句学习。

参考文献

1. Davis M, Logemann G, Loveland D W. A machine program for theorem-proving. Communications of the ACM, 1962, 5(7): 394-397

2. Harrison J. Handbook of practical logic and automated reasoning. Cambridge: Cambridge University Press, 2009, 79-90

（欧阳丹彤　许有军）

Eiffel yuyan

Eiffel 语言（Eiffel language）　一种面向对象程序设计语言。它由美国交互软件公司 B. Meyer 等人于 1985 年设计，1986 年成为软件产品。

基本要素　包括对象、类、继承和实体。

对象　类的实例，在系统运行期间占有一定存储空间。对象的属性域可以是简单的（如数值、字符等），也可以是对其他对象的引用。对象在系统运行期间动态创建与消亡。

类　程序的唯一构造单位，既是模块，也是类型。Eiffel 程序是类的结构化集合，无主程序概念。Eiffel 语言中严格区分了静态的类和动态的对象：类是一组相似对象的抽象正文描述，任何对象都是某个类的实例。类的特征分为属性和程式：属性表示对象状态，程式是作用于对象属性上的操作。特征可以说明为延迟的，其具体实现由后继类给出。含有延迟特征的类称为延迟类。延迟特征虽无实现部分，但其功能可用语言中提供的断言机制进行描述，延迟类的抽象语义性质可由类不变式表述。Eiffel 还提供了类属机制，允许类有表示类型的一个或多个类属参数。

继承　类间的基本关系。Eiffel 语言支持多继承，引入了特征重定义及重命名等设施。

实体　类正文中的标识符，表示在运行时刻的可能引用。实体是比变量更一般的概念，包括程式的局部变量、程式参数、类的属性和用预定义 RESULT 表达的函数结果标识符等。

基本成分　包括类型、表达式、语句。

类型　Eiffel 是强类型语言，允许完全的静态类型检查。每个实体必须有类型说明，并可进行初始化。简单的预定义类型包括 BOOLEAN（布尔型）、CHARACTER（字符型）、INTEGER（整型）和 REAL（实型）。类可用作类型，它是用户自定义类型的设施。为保证同类型的变量在其后继类中保持一致，Eiffel 语言引入了 LIKE（依存类型），即若说明 xLIKEy，则 x 的类型始终保持与 y 的类型相同。

表达式　包括常量、实体、函数调用、CURRENT、带运算符的表达式等。函数调用形式为 x. f(p1, p2, …, pn)，它表示调用与实体 x 相关的对象中的函数 f，pi($i=1,\cdots,n$) 为函数参数。保留字 CURRENT 本身为表达式，它表示类的当前实例。

语句　包括过程调用、赋值、条件、循环、检查、排错等。过程调用与函数调用形式相同。

实现环境　目前 Eiffel 可在 UNIX 版本下运行，并正被移植到其他环境。Eiffel 环境提供了两个编译命令：EC 和 ES，分别用于编译单个类和整个系统。编译程序以 C 作为中间代码，提供了丰富的编译开关，用以控制系统的运行模式。它还提供虚拟存储机制，支持自动存储管理。Eiffel 的基本类库提供了大量描述常用数据类型的类，如数组、表、堆栈和树等。此外，它还提供了程序调试、文档编制和图形设计等工具。

目前，已有 Eiffel 语言的非盈利国际协会(NICE)，负责 Eiffel 语言的标准化和改进；以及 Eiffel 用户小组，负责 Eiffel 的推广应用。

参考文献

1. 徐家福，等. 对象式程序设计语言. 南京：南京大学出版社，1992

2. Meyer B. Eiffel：the language. Prentice Hall, 1992

（张家重　王志坚）

Erlang yuyan

Erlang 语言(Erlang language)　一种通用的函数式并发程序设计语言。

Erlang 语言的第一版由 Joe Amstrong 等人于 1986 年开发，最初是作为瑞典爱立信(Ericsson)公司的内部开发语言，为了快速开发通信系统原型的目的而设计。1998 年 Erlang 成为开源系统。Erlang 的名称源自丹麦数学家 Agner Krarup Erlang，同时也是 Ericsson Language 的缩写。

Erlang 语言采用无用存区回收、严格求值、动态类型等机制，并且在语言层面对并发编程提供强大的支持。其设计受到 ML、Ada、Modula、Prolog 等语言的影响。Erlang 的顺序部分使用严格求值的函数式编程模式，支持匿名函数、高阶函数、模式匹配(pattern matching)以及常见的数据结构，如元组(tuple)和 list 表等。Erlang 的原生数据类型除了常见的数值类型外，还包括如进程号(Pids)和端口(Ports)等直接支持并发和通信的数据类型。

并发机制是 Erlang 区别于其他通用程序设计语言的显著特征，其设计受 Hoare 的 CSP 模型(communicating sequential processes)影响，使用消息传递机制，不支持共享内存。进程是 Erlang 应用的基本组成部分，它不是操作系统层次的进程或线程，而是由 Erlang 运行时系统管理的一种轻量级进程。据估算 Erlang 进程的最小开销只有约 300 计算机字(word)，因此可以高效地管理千万量级的进程，而不会对系统性能产生显著影响。

Erlang 的进程间通信由异步消息传递完成，进程之间没有任何共享状态。每个进程都有一个“邮箱”，存储已收到但还没有被处理的消息。进程可以使用 receive 原语和模式匹配机制来获取符合条件的消息，Erlang 运行时系统提供专门的消息处理功能来帮助进程实现消息的提取。在 Erlang 中，一个消息可以包含所有合法的 Erlang 数据结构，包括原生数据类型(如数值、字符等)和复合数据类型(元组、list 表等)，甚至可以是一个函数。

Erlang 的另一个显著特点是支持代码的热加载(hot code loading)。Erlang 使用模块(module)作为代码的封装和编译单元，进程可以在运行时强制加载模块的最新编译版本，而不需要重启整个系统。

自 1998 年开源以后，Erlang 已成为开放电信平台 OTP(open telecom platform)的重要组成部分，被众多公司和开源项目使用，如 Amazon 的分布式数据库 SimpleDB，GitHub 代码管理平台，ejabbered 即时通信服务平台，Twitter 的 Twitterfall 服务等。另外，在金融领域，Erlang 等函数式语言也越来越流行，高盛公司(Goldman Sachs)就曾经使用 Erlang 开发过高频交易系统。

参考文献

1. Armstrong J. A history of Erlang. Proceedings of the third ACM SIGPLAN Conference on History of Programming Languages, 2007

2. Thompson S, Cesarini F. Erlang Programming：A Concurrent Approach to Software Development. O'Reilly Media, Inc, June 2009

（张昱）

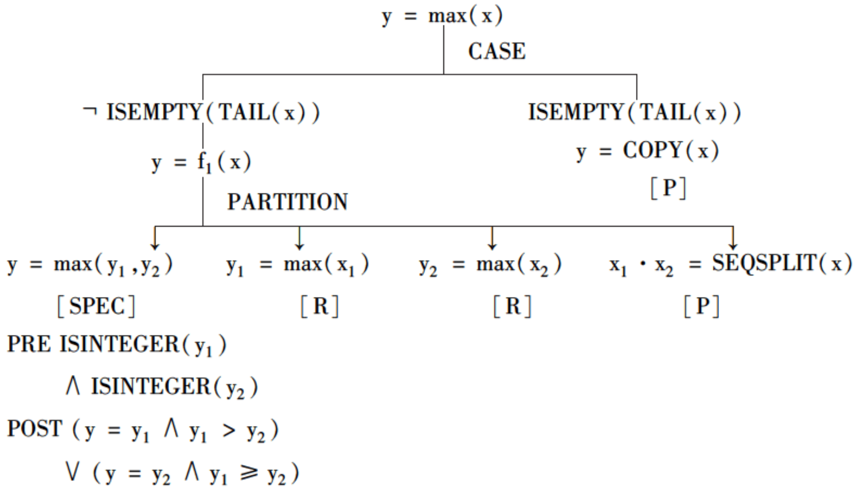
FGSPEC yuyan

FGSPEC 语言(FGSPEC language)　一种基于功能分解的形式化功能规约语言。FGSPEC 用函数作为软件的数学模型，从给定的函数功能规约开始，按某一控制结构将它分解成若干子函数功能规约，直到子函数可用抽象数据类型上的运算实现为止。这样，软件**功能规约**、**设计规约**及其间的转换统一在树形结构上。

FGSPEC 是 functional graphical specification 的缩略语。它是 20 世纪 80 年代后期由南京大学计算机软件研究所在徐家福教授主持下开发的。主要目标是研究从功能规约到设计规约的自动转换。它已于 1988 年在 SUN Sparc 工作站上实现。

FGSPEC 用**前后断言方法**描述功能规约，基本控制结构有 PARTITION 和 CASE。前者将一个功能分解成若干个相关的子功能，其间的相关性由相关变量确定；CASE 则提供条件选择机制，为保证最终程序的确定性，各分支上的条件表达式必须互斥，且所有条件之并是永真式。FGSPEC 提供了抽象数据类型定义设施，用以刻画分解树叶结点上的功能操

作。在定义抽象数据类型的公理方面，FGSPEC 允许在等式右方使用 EXIST，ALL，FIND 和 FIND ALL 四个功能描述成分，以增强描述能力。为了直观，它的规约采用树形图表示。下面是用 FGSPEC 写的求整数序列最大元的规约。其中，叶结点下（ ）中的 P 表示语言直接支持的原始结构，R 是递归结构，SPEC 是用前后断言定义的子功能规约。该规约实际上刻画了求整数序列最大元的一个算法。



参考文献

徐家福，等. 软件自动化. 北京：清华大学出版社，1994

（伊波）

Flash donghua

Flash 动画（flash animation） 用矢量图形来描述对象的形状、大小、颜色和位置等信息的交互式动画技术。

传统的动画片段虽然可以用 MPEG、AVI 等文件格式进行压缩，然后在网络上发布，但由于其数据量仍然很大，给实时传输带来了困难。Flash 动画充分利用了矢量图形来表现剧情，既简洁明了，又减少了文件大小，从而利于网络传播。Flash 动画的一个重要应用是在互联网上发布广告。由于采用矢量图形来描述对象的形状、大小、颜色和位置等信息，不仅产生的文件相对较小，而且显示的图形与分辨率无关，这意味着即使把矢量图形放大到整个屏幕，还能保持相同的文件大小且不会影响显示质量。

Flash 动画可在网络上快速加载的另一个原因是“流式内容”。由于用户在观看动画时并非同时看或听一个文件中的每一个字节，即用户是在逐步接收的。例如，当一个人阅读一本书时，一次仅能看一页。因此，当你阅读网络上的书时，你可能希望只阅读开始的几页，而其余的内容在后台以对用户透明的方式下载。如果必须要等待整本书完全下载才能阅读，用户可能会放弃这个站点而点击别处。Flash 的流式功能意味着即便是带有声音、动画和位图的大型文件也可以几乎同步实时放映。

Flash 动画另一个吸引人之处在于它的交互性。Flash 可创建一种由用户控制的体验，而这种体验将直接取决于设计者在 Flash 中嵌入的交互性能。用 Flash 可以创建按钮以显示信息、播放声音、跳到电影中的不同位置以及响应鼠标事件。用 Flash 设计的电影可以按照预先定义的速度放映，也可以遵循观看人员所输入的路径进行放映。Flash 4 通过一个基本的但却很强大的脚本编辑引擎支持 if-and-end 交互，这意味着 Flash 电影可以下列方式放映：“如果按钮 A 按下，则进行动作 1，否则（或者其他情况，例如如果按钮 B、C 或 D 按下），进行动作 2。”所使用的手段越具有吸引力，观众就越投入。除了前面提到的因素，还有一个因素在 Flash 演示效果中发挥了重要作用，那就是声音。声音的效果在很大程度上决定了人对事物的响应。没有它，即便是再强烈的视觉效果仍然显得不足。Flash 允许添加声音效果或者将屏幕上的动作加上音轨，所有这些将为观众带来真正难忘的感受。

Flash 动画是目前网络上最流行的一种交互式动画格式，这种动画通常需要用 Macromedia 公司开发的 Flash Player 播放器才能正常观看。Flash 动画最初是用 Future Wave 公司开发的 Flash 软件来制作的，该公司被 Macromedia 收购后得以大力发展，而 Macromedia 开发的 Flash 软件成为制作 Flash 动画的主要工具之一。现 Macromedia 已被 Adobe 公司收购。

参考文献

1. 美国 Adobe 公司. Adobe Flash CS4 中文版经

典教程. 陈宗斌,译. 北京:人民邮电出版社,2009

2. http://en.wikipedia.org/wiki/Flash_animation

(全小刚)

FORTRAN yuyan

FORTRAN 语言(FORTRAN language) 一种面向过程的程序设计语言。FORTRAN 是 formula translation(公式翻译)的缩略语。

FORTRAN 语言是在 20 世纪 50 年代中期由美国 IBM 公司的 J. Backus 领导的小组为 IBM704 计算机设计的。第一个 FORTRAN 语言标准称为 FORTRAN 66,在 70 年代修订为 FORTRAN 77,分全集和子集。1991 年国际标准化组织又批准新的 FORTRAN 标准,称为 FORTRAN 90(规定 F 后面的 6 个字母用小写)。它是国际上第一个支持多字节字符集的标准,该标准采纳了我国 FORTRAN 工作组关于 CHARACTER(KIND = ,…)的建议。

FORTRAN 语言包括常数、变量、数组、算术表达式、逻辑表达式等,语句分成赋值语句、输入输出语句和格式语句、控制语句、说明语句以及子程序等几类,详细内容请参见语言标准文本和有关系统的语言基准手册。

FORTRAN 语言主要用于数值计算,由于 IBM 公司在 FORTRAN 语言诞生不久,就为计算机配置了 FORTRAN 编译程序,别的厂商也如此,所以使 FORTRAN 很快普及,又由于 FORTRAN 语言的标准化工作更促进它的推广应用。FORTRAN 语言的特点是接近数学公式,简单易用,功能逐步扩大,如允许复型与双精度浮点运算,子程序定义机制,输入输出的格式说明,允许布尔表达式,函数和子例程名可以作为参数传递。FORTRAN 77 扩充了字符处理功能,使之能应用于非数值运算领域,还增加了块 IF 语句,ELSE 语句和 END IF 语句等,使写出的程序趋于结构化,易读性强。

Fortran 90 又对 FORTRAN 77 作了许多扩充和改进。

(1) 数组运算机制;

(2) 改善了数值计算;

(3) 数据类型参数化,允许使用多种字符类型,满足各国字符处理的需要;

(4) 从 6 种内部数据类型中派生出用户定义的数据类型;

(5) 模块化数据与过程定义机制,提供了一种数据与过程包装的强有力的而又安全的形式;

(6) 指针机制,允许创建和操作动态数据结构;

(7) 增加自由形式的源程序形式;

(8) 提供了过程的递归调用机制;

(9) 提供了附加的控制结构,如 **do... end do**, **do while** <Condition> 等。

由此可以看出,Fortran90 已经是具有强大数值计算能力的现代高级语言,程序的书写更趋结构化,模块化。

现在,国际 FORTRAN 工作组又提出为适应 ISO/IEC 10646 的颁布,要制定更新的 FORTRAN 国际标准。另外,随着计算机科学技术的飞速发展,超级计算机已经进入了向量处理和并行处理时期。作为科学计算的主流程序设计语言,扩充 FORTRAN 90 使之提供向量和并行处理功能,已是其发展的主要趋势,高性能 FORTRAN(HPF)正在设计中。HPF 的目标是支持并行程序设计;能在 SIMD 或 MIMD 机上获得较高性能;便于在不同体系结构的计算机间移植其目标代码,主要扩充数据分布特性和并行语句。

参考文献

1. 全国信息技术标准化技术委员会程序设计语言分技术委员会 FORTRAN 工作组. 标准 FORTRAN 90 语言程序设计. 北京:学苑出版社,1994

2. Adams J C, Brainerd W S, Martin J C, et al. FORTRAN 90 handbook. New York: McGraw Hill,1992

3. Brainerd W S. The programming language standards scene, ten years on FORTRAN. Computer Standards and Interfaces, 1994,16(5 and 6): 459,464

(程虎)

FP yuyan

FP 语言(FP language) 一种无副作用的具有组合子风格的**函数式程序设计语言**。它是由 John Backus 在 1977 年接受 ACM Turing 奖的讲演中提出的。FP 程序是没有变量的函数。FP 表达式的计值是在函数一级的运算,它将函数型(又称组合型)施于函数以产生新的函数,FP 的算子具有较强的代数性质,因此可把程序作为代数项处理。下面的 FP 程序计算二向量的内积:

$$\text{Def IP} \equiv (/+)\circ(\alpha\times)\circ\text{Trans}$$

其中 /, ∘和 α 分别表示组合型“插入”,“复合”和“作用于所有”,它们将已知函数组合成新函数。FP 的特点是:

(1) 引用透明,程序紧凑;

(2) 便于表示递归函数,因而,具有较好的表达可计算函数的能力;

(3) 具有潜在的并行性,便于表示并行算法;

(4) 具有良好的代数性质,因而,便于使用机械方法来理解和证明程序。FP 有多种实现和扩充,最典型的是 J. Backus 后来提出的 FL 语言,它是强类型语言并允许高阶函数和用户自定义类型。

参考文献

Backus J. Can programming be liberated from the Von Neumann style? A functional style and its algebra of program. CACM,1978,21(8):613-641

(黄林鹏　孙永强)

GSPEC yuyan

GSPEC 语言(GSPEC language)　图形化的设计规约语言。将软件功能树形分解和抽象数据类型有机结合起来,用于描述形式软件设计规约。GSPEC 是 graphical specification 的缩略语,由南京大学计算机软件研究所徐家福教授等人于 1987 年提出。

GSPEC 由**抽象数据类型**机制和软件功能分解机制两部分组成。抽象数据类型机制包括抽象数据类型定义机制(DTYDE)和数据类型实例化机制(INSTANTIATION)。数据类型实例化可用于有参抽象数据类型定义和语言中定义的复合类型的例化。抽象数据类型定义机制由基于子句(BASED ON)、引进子句(INTRODUCE)、数据说明子句(DECLARE)、公理子句(AXIOM)组成。基于子句用于刻画多个数据类型之间的层次式依赖关系,引进子句刻画抽象数据类型中所定义操作的语法形式,即型构,数据说明子句对公理子句中出现的所有量作类型说明。公理子句用代数和一阶谓词相结合的方法,通过操作的性质和各操作间的关系来刻画抽象数据类型中所定义的操作的含义。每条公理的一般形式为右部可以带有条件的等式,并且等式右部还可以出现 EXIST(存在量词)、ALL(全称量词)、FIND(找出满足后面公式的任一值),和 FIND ALL(找出满足后面公式的所有值的集合)这四种功能性描述成分。

GSPEC 语言的软件功能分解机制提供了任意有限多叉的树形分解模式,形象直观,层次分明。该语言成分有三种:基本控制结构,控制结构定义和函数定义。基本控制结构有 PARTITION 和 CASE。控制结构定义由控制结构分解树(STRUCTURE)和结构语法子句(SYNTAX)两部分组成,前者刻画该控制结构是如何由其他已知的函数和控制结构组合而成,后者则给出该控制结构的调用方式。函数定义用于完整地刻画软件功能分解的树形结构,通过使用语言提供的基本控制结构和用户定义的控制结构,逐层将描述软件功能的函数分解为子函数,形成一个树形结构,其叶结点函数可以是:语言定义的基本函数,用户用抽象数据类型定义机制刻画的操作,用函数定义机制定义的函数,用其他语言书写的函数,递归调用结点,尚未定义而需要在执行中模拟的函数等。

GSPEC 语言通过在函数分解树的叶结点中使用抽象数据类型中定义的操作,使语言的两种机制有机地结合起来,针对具体问题,用户可有所侧重。语言的图形化表示形象易读,便于软件的理解和维护,语言关于接口的精确刻画及关于抽象数据类型的正确性(终止性,一致性和完备性)验证设施有利于保证和验证软件规约的正确性。

GSPEC 语言已在 SUN 工作站和微型计算机上实现,并开发出若干具有实际意义的中小型软件,例如新构造找水系统等。

参考文献

徐家福,等. 软件自动化. 北京: 清华大学出版社,1994

(费宗铭)

Haskell yuyan

Haskell 语言(Haskell language)　一种通用、非严格语义(non-strict semantics,指按需调用,惰性计算)和强静态确定类型的纯函数式编程语言。它以逻辑学家 Haskell Curry 命名。

1987 年 9 月,在函数式编程语言和计算机系统结构学术年会(FPCA)上,一些学者形成强烈的共识:成立一个委员会来为一个新的集大成的函数式编程语言定义开放标准,作为将来研究函数式编程语言设计的基础。1990 年 4 月,《Haskell 报告》第一个版本问世。随后 Haskell 语言的设计不断完善,Haskell 2010 已经发布。

表领悟(list comprehension)　Haskell 等少数语言特有的语法构造。Haskell 将表作为一个基本概念,并提供简单而强大的表示方式,使得可以从现有的若干表中,通过选择和过滤操作来构造新表。这样,表上的很多常用函数可以清晰而简洁地定义,无须显式的递归。

惰性求值(lazy evaluation)　Haskell 语言的一个重要特征。它是指计算只有在其结果真正需要时才执行,以避免不必要的计算。惰性计算保证了只要有可能程序就会终止,它还允许处理无限表这样带无数个元素的数据结构。例如,利用惰性计算,很容易写出筛法求素数的函数,它不是产生完整的从 2 开始的整数表后再来筛出素数,而是在逐个识别素数过程中需要下一个整数时去获取该整数。

类型类(type class)　作为一种类型构造,首先出现在 Haskell 中。类型类通过约束参数多态性的类型变量来支持特定多态性。这样的约束涉及一个类型变量 a 和一个类型类 T,a 只能实例化到这样的类型,它支持与 T 相关联的重载运算。类型类最初设想作为实现重载的算术和相等运算的一种途径,后来发现它还有很多其他应用。

单子(monad)　Haskell 用来掌控副作用而不破坏纯函数性的一个统一框架,它源于范畴论中的单子概念。Haskell 的单子是一种抽象数据类型构造符,它封装程序基理而不是数据。用函数式风格写的程序利用单子可以构造包括序列化运算的过程或者定义一些任意的控制流,如处理并行性、后续(continuations)、副作用(如输入/输出)或异常的控制流。

在实际系统开发中,Haskell 语言的应用日益广泛。例如,X 视窗系统的视窗管理程序 Xmonad 全部用 Haskell 开发;在可实际应用的操作系统内核 seL4 的验证工作中,seL4 的 API 是用 Haskell 语言写的可执行规格说明表示的。

参考文献

Jones S P. Haskell 98 Language and Libraries: The Revised Report. Cambridge University Press, 2003

(陈意云)

Hopfield wangluo

Hopfield 网络(Hopfield network)　一种具有全互联结构的单层反馈型**人工神经网络**模型,因其由美国加州工学院生物物理学家霍普菲尔德(J. J. Hopfield)提出而命名。Hopfield 网络的输出通过神经元间的反馈连接不断回授到输入端,在输入的激励下,由神经元兴奋机制和神经元间连接强度所决定的神经元状态发生持续不断的变化,使得 Hopfield 网络表现出丰富的动力学行为。

从 20 世纪 40 年代初,人们就认识到:使用理想的神经元连接组成的人工神经网络具有强大的联想、存储、学习与优化功能。众多学者在此基础上开始研究各种有意义的神经网络模型。1982 年,美国加州理工学院的生物物理学家 J. J. Hopfield 提出了一种能够实现联想记忆和神经优化的全互联型反馈神经网络——Hopfield 网络。1985 年,Hopfield 等人给出了对称 Hopfield 网络的稳定性判据,并用来解决约束优化问题,成功求解了计算复杂度为 NP 难的旅行商问题(travelling salesman problem,TSP),之后他还利用多元 Hopfield 网络的多吸引子及其吸引域,实现了信息的联想记忆功能。自 20 世纪 80 年代开始,Hopfield 反馈神经网络已经成功地应用于包括图像处理、语音处理、数据查询、容错计算、模式识别、自动控制等在内的多个工程领域,现在仍不断有一些新应用的报导。

Hopfield 网络模型不仅在理论上能用 Lyapunov 函数分析判别其动态稳定性,而且与电子模拟线路之间存在着明显的对应关系,因此它不仅易于理解,而且便于实现,与其他人工神经网络模型相比,具有简单、快速收敛的显著优点。

Hopfield 网络具有两类最为典型的应用:神经优化(neural optimization)和联想记忆(associative memory)。定义一个对应网络状态的能量函数,若在能量函数最小时 Hopfield 网络能达到收敛平衡,则可将优化和联想记忆问题归结为一个求网络能量函数极小值的问题。在函数优化问题中,将能量函数作为待优化的目标函数,网络的状态变化可看作一个递归优化求解的过程,当网络稳定平衡后,收敛网络的神经元参数中即确定了优化解。在联想记忆中,将网络的收敛态对应为待记忆模式,当给网络一个适当刺激时,网络即可通过演化收敛形成一个动态的分布式联想记忆模型。

根据 Hopfield 网络激活函数的选取不同,可以将其分为离散型的 Hopfield 反馈神经网络(discrete Hopfield neural network,DHNN)和连续型的 Hopfield 反馈神经网络(continuous Hopfield neural network,CHNN)。DHNN 的激活函数为二值型的,其输入、输出取值为离散的{0,1}的反馈网络,主要用于联想记忆,如 J. J. Hopfield 提出的离散型的 Hopfield 反馈神经网络。连续型 Hopfield 反馈神经网络的激活函数的输入与输出之间的关系为连续可微的单调上升函数,主要用于优化计算。由 K 个神经元组成的连续型 Hopfield 神经网络模型如图 1 所示。其中,神经元特性用具有反馈线路的运算放大器模拟,C_i 为输入电容,R_i 为输入电阻,T_{ij} 为神经元之间的连

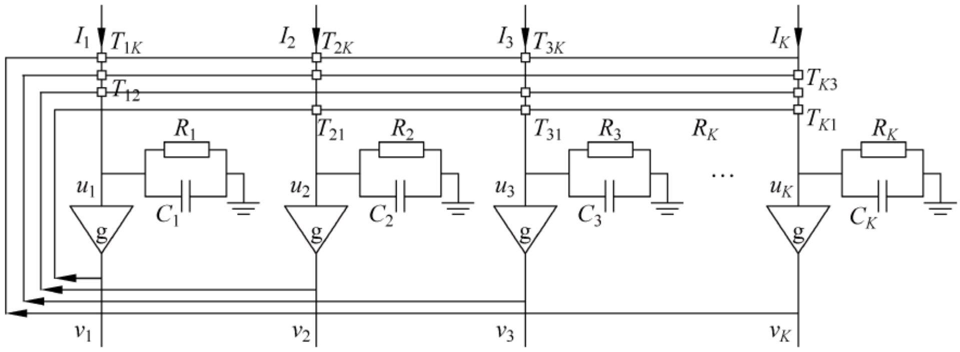


图 1　连续型 Hopfield 神经网络模型

接权值。根据图 1 列出电路方程为

$$\begin{cases} \frac{\mathrm{d}u_i}{\mathrm{d}t} = -\frac{u_i}{\tau_i} + \sum_{j=1}^{K} T_{ij}v_j + I_i \\ v_i = g(u_i) \quad i = 1,2,\cdots,K \end{cases} \tag{1}$$

式中,u_i,v_i,I_i 分别表示第 i 个神经元的输入电压、输出电压和外部输入激励,τ_i 为时间常数,其值等于漏电导 ρ_i 与输入电容 C_i 的乘积,$\rho_i^{-1} = R_i^{-1} + \sum_{j=1}^{K} T_{ij}$。激活函数 $g(x)$ 为连续可微的 Sigmoid 函数,其值为输入 x 与输出的锐度参数 ε 的比值的双曲正切函数值。一般的,将连续型 Hopfield 网络的能量函数定义为

$$E = -\frac{1}{2}\sum_{i=1}^{K}\sum_{\substack{j=1 \\ j\neq i}}^{K} T_{ij}v_iv_j - \sum_{i=1}^{K} I_iv_i + \sum_{i=1}^{K}\frac{1}{\rho}\int_0^{v_i} g^{-1}(v)\mathrm{d}v \tag{2}$$

能量函数中的积分项是人为加上的,它是神经网络电路设计中产生的,在运算放大器的放大倍数足够大时,可以忽略不计,因而它对能量函数和优化问题的结果影响均不大。假设网络是对称的,对式(2)求时间导数,可知能量对时间的导数值恒小于等于 0,即随着时间的变化,网络状态轨迹总是沿着能量函数减小的方向演化,当 $t\to\infty$ 时,网络收敛到稳态。网络的稳态平衡点对应于其计算能量函数的极小点,因而它可以广泛地用于神经优化和联想记忆问题。

但是,目前 Hopfield 网络的样本容量(即网络中能还原的样本个数)比较小的问题始终限制其发展。为了进一步提高 Hopfield 网络的容量,国内外已做了大量的研究工作,但基本上都是以增加样本的维数,扩大网络的规模和复杂程度,以及牺牲网络的收敛速度为代价的。因此,如何充分的保留和发挥 Hopfield 反馈神经网络自身的优势,又进一步提高 Hopfield 反馈神经网络的样本容量是一个非常值得研究的问题。

参考文献

1. Hopfield J J. Neural Networks and physieal systems with emergeni colleetive computational abilities. Proeeeding Nation Aeademy Science USA, 1982,79: 2554-2558

2. 史忠植. 神经网络. 北京: 高等教育出版社,2009

(焦李成)

Internet jiben fuwu

Internet 基本服务(basic services of Internet) Internet 为用户提供的基本服务,主要包括**文件传送**、远程登录和**电子邮件**服务。为提供这些服务,Internet 制定了相应的协议:

(1) 文件传送协议(FTP)是描述客户机与远程主机之间传送文件的协议(参见**文件传送**)。用户和远程主机之间传送文件时,用户必须在远程主机上有合法的用户账号和口令以及相应的访问权限。匿名文件传送协议(anonymous FTP)允许没有对方计算机注册账号和口令的 Internet 用户登录对方计算机,并获取所需的文件。在这种服务器上有一个名为 anonymous 的特殊注册账户,用户用这个名字去登录,用自己的电子邮件地址或本地服务器的提示作为口令去访问,便可以获得公开发布的文件。

(2) 远程登录协议(Telnet)是标准的 Internet 虚拟终端协议,实现客户机远程登录到 Internet 上的某台主机。用户登录到远程主机后,就成为该远程主机的虚拟终端用户,可共享该主机的软、硬件资源和数据库等。

(3) 简单邮件传输协议(SMTP),是描述客户机与远程主机之间传送电子邮件的协议(参见**电子邮件**)。

(胡道元)

IPSec

IPsec(Internet Protocol Security)　IETF 的 IP 安全协议工作组针对 IP 协议设计中存在的安全缺陷而设计的一组安全规程，以保护使用 IP 进行传输的各类高层协议如 TCP 和 UDP 协议。在 IPv6 协议中，IPSec 被定义为必须实现的规程。IPSec 涉及的安全机制包括鉴别、完整性保护、访问控制、保密等，还包括与这些机制相配套的密钥管理协议。

IPSec 在 IP 协议之外增加了两个规程来提供传输的安全：鉴别头(authentication header, AH)和负载安全封装(encapsulating secutiry payload, ESP)。鉴别头规程提供了无连接的完整性保护、数据源鉴别和可选择的防回放服务；负载安全封装规程提供加密和流量填充服务，也可以提供无连接的完整性保护、数据源鉴别和可选择的防回放服务。根据具体的安全需要，这两个规程可以独立使用和组合使用。每个规程都提供了两种工作模式，传输模式和隧道模式，传输模式提供端到端的保护，而隧道模式提供 IP 报文隧道。IPSec 规程文档的关系如图 1 所示。

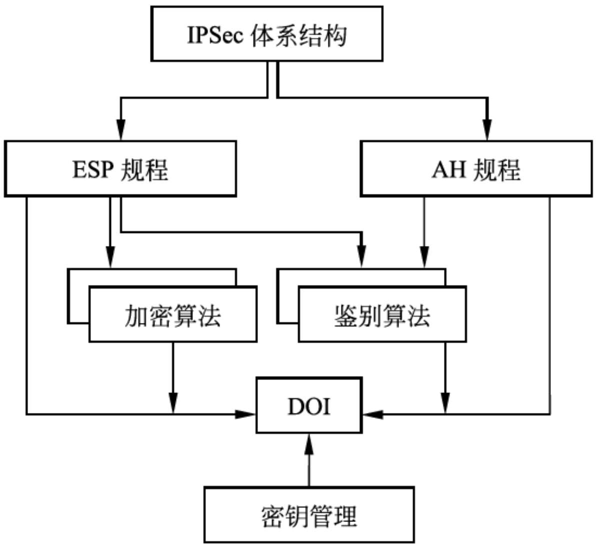


图 1　IPSec 规程文档关系图

表 1　安全协议数据封装格式

协议＼模式	传输模式	隧道模式
AH	IP 头 + AH 头 + 负载数据	IP 头 + AH 头 + IP 头 + 负载数据
ESP	IP 头 + ESP 头 + 负载数据 + ESP 尾	IP 头 + ESP 头 + IP 头 + 负载数据 + ESP 尾
AH-ESP	IP 头 + AH 头 + ESP 头 + 负载数据 + ESP 尾	IP 头 + AH 头 + ESP 头 + IP 头 + 负载数据 + ESP 尾

在 IPSec 中，最重要的概念是安全联系(security association, SA)。SA 表示一个安全的连接，IPSec 通过 SA 的概念在无连接的 IP 服务中引入了一些面向连接的特性，即为特定的通信活动提供安全服务的上下文。SA 的工作方向是单工的，在需要双向通信的点到点连接中，需要提供两个安全联系。SA 由一个唯一的三元组表示：安全参数列表(security parameter index)、IP 宿地址和安全规程(AH 或 ESP)标识。原则上宿地址可以是单播地址、组播地址和广播地址，但目前 SA 机制仅支持单播地址。在传输模式下，SA 用于实现两个主机之间的安全连接，在传输模式下，SA 可以支持安全网关到安全网关和主机到安全网关之间的连接。解释域(domain of interpretation, DOI)定义了负载的格式，支持的安全规程以及对安全相关信息的命名约定，比如对安全策略或者加密算法和模式的命名。

参考文献

龚俭，吴桦，杨望. 计算机网络安全导论. 南京：东南大学出版社，2009

(杨望)

Jackson xitong kaifa fangfa

Jackson 系统开发方法(Jackson system development method)　一种面向数据结构的软件开发方法，该方法是以数据结构为基础，通过一组映射或转换过程来建立程序的结构。

JSD 是 Jackson System Development 的缩写。JSD 方法是由 M. A. 杰克逊提出的。他首先于 1972—1974 年间提出了 JSP，这是一种结构化程序设计方法，后来在 1978—1981 年将其扩充为 JSD。

JSD 方法把系统开发分为描述和实现两个阶段。描述阶段建立一个与系统相关的客观世界的模型，并在此基础上确定系统功能。实现阶段在具体的软硬件环境下实现系统功能。

JSD 方法有 6 个步骤：

(1) 实体动作步骤：列出客观世界中与系统有

关的原始实体和实体动作表以及每一动作的属性表。这些表规定了系统功能的范围。

（2）实体结构步骤：用结构图来表达每一个实体的诸动作间有序约束，得到一个结构图集。结构图中包括顺序、选择及重复等三种构造。

实体动作步骤和实体结构步骤的结果是一个客观世界的抽象描述。为了模拟客观世界，客观世界中的每一个实际进程需要有一个对应的模型进程。

（3）初始模型步骤：说明模型进程是如何与其相应的实际进程进行联系的。JSD 方法支持数据流联结与状态向量联结两种方式，结果得到一个系统说明图。此外，软件开发人员还需写出模型进程的结构文本以便进一步说明模型进程。结构文本实质上是结构图的文本形式，它包含了从外界输入信息的操作，以及根据输入信息确定模型进程中操作执行的条件。

（4）功能步骤：指明系统功能。指明的方式有 3 种：①以初始模型中的术语为基础，用自然语言按规定的形式陈述功能。这种方式有利于软件开发人员与用户之间的通信。②在初始模型的系统说明图中加入功能规约。③用结构文本给出细化的功能规约。这 3 种方式一般要结合使用。

（5）系统时序步骤：在时间上系统总是延迟于客观世界，而根据系统功能的需要，系统的不同部分在延迟的程度上总是有所区别的。为了完善系统规约，在本步骤中考虑这些延迟，并和系统用户一起讨论系统中各个不同部分的延迟程度。结果得到非形式的描述时序约束的文件。

以上 5 个步骤为描述阶段。

（6）实现步骤：在具体的软硬件环境下实现系统功能。实现步骤的基本任务有 3：①确定有多少真实或虚拟的处理器用于执行系统；②当处理器的数量小于进程的数量时，确定如何在进程间分配处理器；③在一个拥有多个进程的处理器上确定如何由多个进程共享处理器时间，即进行进程调度。

描述阶段得到的系统规约图展示了系统中有什么样的进程，进程之间如何联结以及系统边界的输入和输出，而结构图则展示了进程的结构，结构文本又进一步给出了进程的细化规约。实现阶段得到的调度进程指明了在具体环境下系统内部的事件发生的次序，系统实现图展示了系统的实现轮廓。在这些信息结构的指导下最后完成系统代码的编制。

JSD 中的每一步骤的结果作为下一步骤工作的部分根据，但实际上开发并不是顺序地从上一步到下一步，回溯是不可避免的，JSD 的目标是尽量减少这种情况的发生。

JSD 方法是一种成熟的系统开发方法。它的基本特点体现在下面几个方面：①软件开发人员首先描述客观世界，即建立客观世界的模型，然后在模型的基础上提出系统功能。②动态的客观世界需要动态的模型来模拟。JSD 中的模型是用具有动态概念的顺序进程来描述的，因此，JSD 方法很适合于强调时序的应用。③JSD 的实现技术允许软件开发人员在建立系统时自行决定进程的调度，而不是在系统运行时由操作系统来决定。④JSD 方法致力于在开发过程中得到一些数据结构表示，如进程的结构图和结构文本，并以此为基础进行程序设计。

参考文献

Jackson M A. System development. Prentice - Hall, 1983

（郝克刚　葛玮）

Java yuyan

Java 语言（Java language）　一种简捷的、面向对象的、用于网络环境的程序设计语言。Java 语言是由 SUN MicroSystem 公司于 1995 年 5 月正式对外发布的。

Java 语言的基本特征是：简洁易学、面向对象、适用于网络分布环境、解释执行和多线程、具有一定的安全健壮性。

简捷易学——最初开发 Java 语言的本意是为家用电器的程序控制而用的。它坚持面向对象的基本原理，但又避免了运算符重载、多重继承等复杂概念。它基本上是一种解释执行的语言，而且其基本解释程序和对于类的支持只有 40 kB 左右，加上标准类库和线程支持也只有 215 kB 左右，因此，系统开销较小，适用于小型的信息处理和信息环境。它由于能够实现自动废区收集，因此也简化了程序设计的内存管理工作。

面向对象——在坚持面向对象方法的基础上，Java 提供了极简单的类机制，以及很有效的接口模型。Java 的对象中封装了其状态变量和相应方法，实现了模块化和信息隐蔽；而通过类的继承机制，子类可以使用父类所提供的方法，从而实现了代码复用。

适用于网络分布环境——Java 是面向网络应用

的语言,通过它所提供的类库,可以处理 TCP/IP 协议规程,可以通过 URL 地址在网络上访问其他对象,能较方便地与其他计算结点协同工作。

解释执行和多线程——Java 解释程序能直接对 Java 的字节码进行解释执行,由于可从字节码获得部分编译信息,因此使得连接过程更加简捷。Java 所提供的多线程机制可使应用程序并行执行,其同步机制也有助于实现数据共享。

安全健壮——由于 Java 提供了自动废区收集、面向对象的异常处理、自动捕获类型声明中的常见错误、一切对内存的访问都必须通过对象的实例变量实现(不支持指针)等手段,因此 Java 可防止部分故障,具有一定的安全健壮性。

由于 Java 具有以上特性,所以已受到各种应用领域的重视,取得很快的发展,在因特网上已推出了用 Java 语言编写的多种应用程序。但 Java 还不很成熟,尚有许多可改进之处。随着 Java 芯片、Java OS、Java 解释执行和编译、Java 虚拟机技术的日趋先进,Java 语言将更加完善,发挥更大的作用。

参考文献

1. The Java language: an overview. From http://java. sun. com

2. 王克宏,郁欣,等. Java 语言编程技术. 北京:清华大学出版社,1997

3. 王克宏,李京华,等. Java 虚拟机规范. 北京:清华大学出版社,1996　　(汪成为　王克宏)

Larch yuyan

Larch 语言(Larch language)　代数方法和一阶谓词方法相结合的形式软件规约语言簇,它旨在开发出各种技术和工具以支持形式规约的有效使用。Larch 语言由美国麻省理工学院 J. V. Guttag 等人于 1983 年提出。

Larch 形式软件规约语言簇的主要目的是为各类**程序设计语言**的程序模块提供必要的描述手段,考虑到不同的程序设计语言常有不同的**语法**表示和**语义**定义,因而程序模块的规约就与某一特定的程序设计语言有关,为使和程序设计语言有关的特性局部化,Larch 提供了描述各类程序设计语言的程序模块共性的共享语言和描述各个特定程序设计语言个性的接口语言。Larch 语言簇中每一成员由共享语言及某一特定接口语言组成。

Larch 共享语言采用代数方法,其描述单位为 trait,提供一组供接口规约使用的类符和操作,由语法部分和公理部分组成。语法部分刻画了操作符的型构,公理部分以等式的形式给出。每一 trait 定义了一个相关的理论,作为它的语义,区别于常规的以模型作为语义的处理。trait 定义中通过使用 imports和 includes 子句以便从较小的 trait 组合成较大的 trait,通过定义生成操作集和识别操作集扩展所定义的理论。Larch 共享语言的重要特征之一是引入多种语言成分以便于语义检查,trait 中提供了 implies 子句用于表达用户希望相应 trait 的理论所蕴含的定理。

接口语言的目的是从软件规约的角度,将共享语言和特定的程序设计语言有机结合在一起,并提供描述程序模块中相应运算的机制。其中与特定语言有关的成分,如过程名、形式参数、类型、异常处理等描述的语法借助于该语言的语法,而其中过程的功能刻画用基于一阶谓词的前后断言的形式,描述中可以使用共享语言中 trait 的操作和类型。

Larch 语言通过区分共享语言和接口语言,使和程序设计语言有关的特性局部化,其共享语言具有易组合和强调语义检查的特点。共享语言采用代数方法,接口语言基于一阶谓词演算,具有较好的理论基础。

Larch 已基于项重写系统实现,可用于分析其形式规约,检查其中的语义约束。Larch 语言已用于描述诸如图书馆系统等一系列软件规约。

参考文献

Guttag J V, Horning J J. Report on the Larch shared language. Science of Computer Programming, 1986,(6):103-134　　(费宗铭)

Linux caozuo xitong

Linux 操作系统(Linux operating system)

一种和国际上流行的 UNIX 同类的作为自由软件的操作系统。UNIX 是商品软件,而 Linux 则是一种**自由软件**。它是遵循 GNU(该词最初为自由软件倡导者 Richard Stallman 为其操作系统所起之名)组织倡导的通用公共许可证(GPL)规则而开发的,其源代码可以免费向一般公众提供。因此,Linux 多年来在教育界和学术界十分流行,成为广大计算机编程人员学习研究的对象。许多人还对之进行修改补充,以满足特定的需要。

Linux 是由芬兰科学家 Linus Torvalds 于 1991

年编写的一个操作系统内核。当时他还是芬兰赫尔辛基大学计算机系的学生,在学习操作系统课程中,自己动手编写了一个操作系统原型,从此,诞生了一个新的操作系统。Linux 把这个系统放在 Internet 上,允许自由下载,许多人对这个系统进行改进、扩充、完善,并做出了关键性的贡献。Linux 由最初一个人写的原型演化成在 Internet 上由无数志同道合的程序高手参与的一场活动。

Linux 属于自由软件,而操作系统内核是所有其他软件最为基础的支撑环境,再加上 Linux 的出现时间正好是 GNU 工程已完成了大部分操作系统外围软件,水到渠成,可以说 Linux 为 GNU 工程画上了一个圆满句号。除了 Linux 外,还有许多自由软件,如 NetBSD、OpenBSD 等都是较优秀的具有自由版权的 UNIX 类操作系统。Apache 也是一个著名的自由软件,已在服务器上广泛使用,支持包括 Linux, Free BSD, Solaris 及 HP-UX 等很多操作系统平台。

短短几年,Linux 操作系统已得到广泛使用。1998 年,Linux 已在构建 Internet 服务器上超越 Windows NT。计算机的许多大公司如 IBM, Intel, Oracle, Sun 等都大力支持 Linux 操作系统。各种成名软件纷纷移植到 Linux 平台上,运行在 Linux 下的应用软件也越来越多。Linux 的中文版已开发出来,并开始在中国流行。同时,也为发展我国自主操作系统提供了良好条件。

Linux 是一个开放源代码、Unix 类的操作系统。它除继承了历史悠久和技术成熟的 UNIX 操作系统的特点和优点外,还进行了许多改进,成为一个真正的多用户、多任务通用操作系统。1993 年,第一个产品版 Linux1.0 问世时,全部按自由扩散版权进行扩散,即公开源码,不准获利。不久发现这种纯粹理想化的自由软件会阻碍 Linux 的扩散和发展,特别限制了商业公司参与并提供技术支持的积极性。于是 Linux 转向通用公共许可证(GPL)版权,除允许持有自由软件的各项许可权外,还允许用户出售自由软件拷贝程序。这一版权上的转变后来证明对 Linux 的进一步发展十分重要。

从某种意义上来说,Linux 是 Unix 和 Internet 国际互联网结合的产物。自由软件 Linux 是一个充满生机、已有巨大用户群和广泛应用领域的操作系统,它是唯一能与 UNIX 和 Windows 较量和抗衡的一种操作系统。

从技术上讲,Linux 具有如下特点:①继承了 UNIX 的优点,又有许多改进,能紧跟技术发展潮流,具有很强的生命力;②通用的操作系统,可作为 Internet 上的服务器,可用作网关路由器,可用作文件和打印服务器,也可供个人使用;③内置通信联网功能,可让异种机联网;④开放的源代码,有利于发展各种特色的操作系统;⑤符合 POSIX 标准,各种 Unix 应用可方便地移植到 Linux 下;⑥提供庞大的管理功能和远程管理功能;⑦支持大量外围设备;⑧支持 32 种文件系统,如 Ext 2、Ext、Xiafs、Isofs、Hpfs、MSDOS、UMSDOS、Proc、NFS、SYSV、Minix、SMB、Ufs、Ncp、VFAT、AFFS 等;⑨提供图形用户接口(GUI),有图形接口 X-Windows,有多种窗口管理器;⑩支持并行处理和实时处理,能充分发挥硬件性能;⑪可自由获取源代码,在 Linux 平台上以低成本开发软件。

参考文献

1. http://freesoft.cei.gov.cn

2. http://www.linux.org

3. 汤荷美,董渊,李莉,等. Linux 基础教程(1)操作系统基础. 北京:清华大学出版社,2001

(周锡令　费翔林)

LISP yuyan

LISP 语言(LISP language)　**一种表处理语言**。LISP 是 list processing 的缩略语,它是由 John McCarthy 于 20 世纪 50 年代后期提出的,其理论基础是 λ 演算。LISP 语言的程序由一些函数组成。函数的构造和数学上递归函数的构造方法类似,即从几个基本函数出发,通过一定的方法构造出新的函数。在 LISP 中常用的关于表处理的函数有 cons, car, cdr, null 等。下面的 LISP 程序定义了一个计算阶乘的函数 fac:

```
(define fac (n)
   (if ( = n 0)
   1
   ( * n(fac( - n 1)))))
```

LISP 语言的主要特点有:

(1) 使用条件表达式书写递归函数;

(2) 使用表数据结构和高阶函数进行编程;

(3) 使用 S-表达式表示数据结构;

(4) 在实现上使用 cons 结构和无用单元回收策略进行存储管理。

LISP 是最早的**函数式程序设计语言**之一。它有许多不同的实现和扩充,典型的有 MACLISP、In-

terLISP、EuLISP 等,其中最有影响的是 Common LISP,另外,有与面向对象技术结合的 CLOS 语言,并行语言 StarLISP,以及称为现代 LISP 的 Scheme 等。

LISP 已广泛用于**人工智能**研究的各个领域。

参考文献

1. McCarthy J. Recursive Functions of Symbolic Expressions and their Computation by Machine. CACM,1960,3(4)

2. Steel Jr G L. Common Lisp the Language. Ditital Press,1984

(黄林鹏　孙永强)

LR(*k*) wenfa

LR(*k*)文法(LR(*k*) grammar)　通过查看位于当前输入位置右边的 *k* 个输入符号,就能完成分析的文法。这里,LR 指从左向右,*k* 代表查看 *k* 个输入符号。LR(*k*)文法,是一类上下文无关文法,我们可为它构造无回溯"有效"的分析器。所谓"有效"是指在处理输入长度为 *n* 的字符串的过程中,能够在 C_1n 时间与 C_2n 空间内完成分析,其中,C_1,C_2 为两个常数。

一般地说,大多数用上下文无关文法描述的程序语言都可用 LR 分析器予以识别。LR 分析法比算符优先分析法或其他的"移进-归约"技术应用得更加广泛,而且识别效率甚佳。比普通的无回溯的自上而下方法也要好。LR 分析法在从左至右扫描输入串时,就能发现其中的语法错误,并能准确地指出出错位置。

这种分析法的一个主要缺点是,若用手工构造分析程序则工作量相当大。因此,必须求助于自动产生这种分析程序的产生器。这种产生器称为 LR 分析程序自动产生器。应用这种产生器,能自动产生一大类上下文无关文法的 LR 分析程序。这种产生器还能指出文法有歧义的情形或难于分析的特殊结构。

构造 LR 分析器的主要任务是构造分析表。分析表恰好识别文法所产生的全部语句。分析表的构造通常有四种方法。第一种,也是最简单的一种,叫做 LR(0)表构造法。这种方法的局限性极大,但它是建立其他较一般的 LR 分析法的基础。第二种叫做简单 LR(简称 SLR)表构造法。虽然,有一些文法构造不出 SLR 分析表,但是这种构造法是一种容易实现又极有使用价值的方法。第三种叫做规范 LR 表构造法,这种分析表能力最强,适用一大类文法,但实现代价过高,分析表的体积太大。第四种叫做向前 LR 表构造法(简称 LALR)。这种分析表的能力介于 SLR 和规范 LR 之间,稍加努力,就可以高效地实现。

LR 分析法由 D. E. Knuth 于 1965 年首先提出,1969 年,De Remer 把这个方法简化成实际可用的算法,即 SLR 和 LALR 算法。给定 LR(1)系统,优化过程实质上将导出 LALR 分析器。同时,LR(0)文法也可转化为 SLR 系统。但并不是所有 LR 文法,这种优化均可以做到。因此,以这种方式优化的语言集比用 LR 文法系统表示的语言集小得多。但对多数程序语言来说,用 SLR 分析也就足够了。

参考文献

陈火旺,钱家骅,孙永强. 程序设计语言编译原理. 2 版. 北京:国防工业出版社,1984

(陈火旺　贲可荣)

MapReduce

MapReduce(映射/归约并行编程模型)　一种用于海量数据处理的并行软件编程模型,程序员负责编写应用层的映射(map)函数以及归约(reduce)函数,程序动态并行执行,负载均衡,容错则均由运行时系统负责。系统首先对输入数据进行划分,传送给映射函数一系列键/值(key/value)对,映射函数生成零个或多个中间键/值(key/value)对,并由系统重新进行调度,输入给归约函数做进一步处理,MapReduce 这种高层次面向算法的编程模型简化了开发并行应用的复杂性,适用于并行海量数据处理,例如检索、数据挖掘、机器学习等。MapReduce 由 Google 提出并基于 C ++ 实现,现已在集群、多核和 GPU 等多种平台上均有实现,另一重要开源实现 Hadoop 由 Apache 基金会基于 Java 开发。工业界还设计了许多类似的约束型编程模型,包括微软基于有向无环图(DAG)的任务计算模型 Dryad,Google 图批量同步处理系统 Pregel 和增量式计算框架 Percolator,Yahoo 的数据流计算系统 S4 等。

参考文献

Dean J, Ghemawat S. MapReduce: simplified data processing on large clusters. Commun. ACM, 2008, 51(1): 107-113

(袁良　张云泉)

Miranda yuyan

Miranda 语言(Miranda language)　一种非

严格的纯**函数式程序设计语言**。它是 1985 年至 1986 年间由 D. Turner 设计的。Miranda 的程序为一组递归定义的函数和其他数据对象。Miranda 使用卫士式等式或模式匹配表示情况分析。一个基于卫士式等式的阶乘函数可写成

```
fac x  = 1                x = 0
       = x * fac(x - 1)   otherwise
```

而一个基于模式匹配的阶乘函数可表示为

```
fac 0 = 1
fac(x + 1) = (x + 1) * fac(x)
```

在实现时,前者的控制特征是显式的,而后者是隐式的。Miranda 是第一个被广泛传播的具有惰性计值语义和多态强类型的函数式程序设计语言。它已运行于世界上 600 多个场所,包括在 250 所左右的大学用于教学。Miranda 系统工作于 UNIX 环境下,是 Research Software Limited 公司的商业产品。

[注] Miranda 是 Research Software Limited 公司的注册商标。

参考文献

1. Turner D. Miranda: A non-strict functional language with polymorphic types. Proceedings FPCA'85, Springer LNCS, 201: 1-16

2. Turner D. An overview of Miranda. SIGPLAN Notices, 1986, 12(12): 158-166 (黄林鹏 孙永强)

ML yuyan

ML 语言(ML language) 一种严格的**函数式程序设计语言**。ML 是 metalanguage 的缩略语,它是 20 世纪 70 年代中期由 R. Milner 提出的,当时的目的是作为由 M. J. Gordon 等人开发的称为 LCF 系统的命令语言,该系统用于证明称为 PPλ 的程序理论的性质。ML 是一种强多态类型的语言,一个 ML 程序是一个包含变量定义和函数作用的表达式序列。ML 虽被称为是函数式语言,但它具有引用(即变量可以赋值)的概念并且它的 I/O 系统也引入了副作用。

1984 年,一个由爱丁堡大学、剑桥大学、贝尔实验室和 INRIA 组成的委员会在 R. Milner 的领导下对 ML 语言进行标准化工作,最后的结果加进由 D. MacQueen设计的模块机制并被称为 SML(Standard ML)。SML 是一个比较完善和实用的函数式程序设计语言,它有高阶函数功能、I/O 机制、参数化的模块系统和完善的类型系统。一个简单的例子是:

```
let fun sum i tot = if i = 0 then tot
    else sum(i - 1) (tot + i)
in sum 10 0
end
```

它计算 1 + 2 + 3 + ··· + 10 的值。

ML 语言的其他变形有 Lazy ML, CAML, CAML-Light 等。

参考文献

1. Milner R, et al. The definition of standard ML. Revised ed. MIT Press, 1997

2. Milner R, Tofte M. Commentary on standard ML. MIT Press, 1991 (黄林鹏 孙永强)

Modula-2 yuyan

Modula-2 语言(Modula-2 language) 一种**程序设计语言**。它直接源于 Modula,二者均由 N. Wirth 于 20 世纪 80 年代初设计,它们继承了 **PASCAL 语言**中良好的传统构造,其中包括典型的控制结构、数据类型和过程等概念。并弥补了 PASCAL 语言的不足,增加新的设施,即引进了模块和进程概念,增加低级设施,采用更为系统化的语法等。

其主要特点是:

(1) 模块性 在 Modula-2 中,把模块分成两个语法成分,即定义性模块和实现性模块,并把那些在模块外可见的对象列在一个明显的移出表中,在模块内引用模块外的对象列在移入表中。Modula-2 的模块设施有利于实现模块的分别编译。

(2) 良好的控制结构 每个控制结构都以关键字结尾,可避免歧义性,减少不必要的"BEGIN"。提供了丰富的循环控制结构。特别引进 LOOP 和 EXIT 语句,可以较方便地代替 GOTO 语句。

(3) 输入输出功能由一组模块来实现,这组模块的层次结构反映了输入输出功能的多种抽象级别。

(4) 与机器和实现有关的低级设施放在伪模块 SYSTEM 中,可在高级语言级上实现。

(5) 提供一个层次较高的模块 Processes 和协同程序,有利于实现并发处理。

目前,世界上已经开发了近百个 Modula-2 编译系统。欧洲、加拿大、澳大利亚等不少大学已经用 Modula-2 代替 PASCAL 语言作为计算机科学系本

科生的第一门程序设计课。

1984 年英国标准化学会开始进行 Modula - 2 标准化工作,国际标准化工作始于 1987 年(ISO /IEC JIC1 /SC 22 WG 13),标准化的一个新颖和重要的方面是首次采用形式化方法,用形式化定义(VDM - SL 和扩充的 BNF)来表达语言各成分的**语法**和**语义**。并伴以自然语言规约语句和注释。

参考文献

1. Wirth N. Programming in Modula - 2. 2nd ed. New York: Springer - Verlag, 1983

2. 郑国梁. Modula - 2: PASCAL 的直接后继语言. 微型计算机, 1987, (3)

(郑国梁)

NP lei wenti

NP 类问题(class NP of problems) **计算复杂性理论**中一类重要问题,即多项式时间可验证的问题。

在计算复杂性理论中,问题由无数具体的实例组成,每个实例经编码可表示成一个字符串。所有具有肯定回答的实例所对应的字符串组成一个语言(参见 **P 类问题**)。

图灵机由一个有限控制器、一条输入带和相应的读写头组成。图灵机根据有限控制器的状态和读写头所读到的字符,将执行以下三个操作:读写头在其所在方格内写下一个字符,改变机器状态,最后读写头向左(或向右)移动一格。如果对于任意一个状态和一个字符,要执行的动作(包括上述三个操作)都是唯一的,这种图灵机叫做**确定型图灵机**。如果要执行的动作有有穷多个可供选择,那么此时从第一步到最后一步,图灵机就有许多种可能的动作序列。如果规定,对给定的输入字符串 w,所有这些动作序列中只要有一种导致接受状态,机器就接受 w,这种图灵机就称为**非确定型图灵机**。

一个语言 L 称为是属于语言复杂性类 NP,如果存在一个非确定型图灵机 M 和一个多项式 p,对于每个长度为 n 的字符串 w,且 w 是 L 的字符串,M 都能在 $p(n)$ 步内停机,并接受 w。

一个问题称为是 NP 类问题,如果它对应的语言属于 NP 类。从定义可以看出,P 类问题都是 NP 类问题,且 NP 类问题似乎要比 P 类问题更广。实际上,假定非确定型图灵机在每一步,都恰有两个动作可供选择。此时,要确定一个字符串是否被接受(即问题是否有肯定回答),图灵机在 $p(n)$ 步内,要考察 $2^{p(n)}$ 种不同的序列(这需要指数时间!)。现实中,许多问题正是以这种方式判断一个实例是否有肯定回答的,它们都是 NP 类问题,看上去很难有多项式时间算法。但 NP 是否真比 P 大,至今尚无定论。

非确定型图灵机还有另一种模型(见图 1)。它由一个确定型图灵机和一个猜测模块组成。对于给定输入字符串 w,首先由猜测模块在带上 −1 格及其左方任意写下(猜测)一个字符串,然后转由有限控制器控制,按确定型图灵机方式工作。由于猜测的任意性,机器可能有许多种动作序列,只要其中一种导致接受状态,机器就接受 w。已经证明,以上两种非确定型图灵机模型彼此等价。正因为这个模型,NP 类问题被看成多项式时间可验证问题。

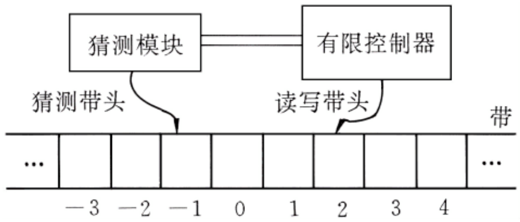


图 1　非确定型图灵机的另一种模型

参考文献

Garey M R, Johnson D S. Computers and intrac - tability. San Francisco, CA: Freeman, 1979

(徐美瑞)

NP wanquan wenti

NP 完全问题(NP complete problem) **计算复杂性理论**中的一类重要问题,其中每个问题都是 **NP 类问题**,但是否任何一个 NP 完全问题都是 **P 类问题**,目前尚无定论。NP 完全问题的研究,在理论上和实践中都有重要意义。

假定给了两个问题 q 和 q_0,如果存在一个确定型**图灵机** M 和一个多项式 p,对于 q 中任意一个长度为 n 的问题实例 x,M 都能在 $p(n)$ 时间内,计算出 q_0 的一个实例 y,使得 x 是 q 中有肯定回答的实例,当且仅当 y 是 q_0 中有肯定回答的实例,就称问题 q 多项式时间多一归约到问题 q_0。显然,如果有这样的归约,q 的求解问题就归结为 q_0 求解的问题。

对于一个问题 q,如果 q 属于 NP 类,且 NP 中任意一个问题都可以多项式时间多一归约到 q,就称 q 为 NP 完全问题。在现实生活中存在大量 NP 完全问题,它们分布在计算机科学、数学、逻辑学和运筹学等许多学科领域,总数已达数千。下面是几个典

型的 NP 完全问题。

（1）**旅行商问题**：给定 n 个城镇和一个界限 B，以及城镇间的距离 $d_{i,j}(1 \leqslant i,j \leqslant n)$，问是否有一条旅行路线，经过每个城镇一次且仅一次，最后返回出发城镇，且其总路程不超过 B。

（2）**划分问题**：给定 n 个自然数，是否能将它们分成两部分，使得两部分自然数各自的和数彼此相等。

（3）**可满足性问题**：对于任意给定的由“与”，“或”，“非”和逻辑变元组成的布尔表达式，是否有对式中各变元的一组赋值，使该表达式的值为真。

（4）**带优先次序的调度问题**：有 m 个处理机和一个任务集合，每个任务所需时间为 1，已知任务间的优先次序（任务间不一定都有次序）和一个截止时间 D。问是否有一个 m 个处理机的调度方法，满足优先次序要求，且在截止时间 D 以前完成全部任务。

由于 NP 类中任意一个问题都可以多项式多一归约到 NP 完全问题，因此，直观上人们可以认为 NP 完全问题是 NP 类中“最难的”问题。对于 NP 完全问题，一般都有一个明显的指数时间算法，但按目前研究结果，要找到一个现实可计算的算法，即多项式时间算法，却十分困难，甚至很可能根本不存在这样的算法（参见 **NP 完全性理论**）。NP 完全问题的求解，只有另辟蹊径，例如寻找多项式时间近似解法。

参考文献

Garey M R，Johnson D S. 计算机和难解性：NP 完全性理论导引. 张立昂，沈泓，毕源章，译. 北京：科学出版社，1987

（徐美瑞）

NP wanquanxing lilun

NP 完全性理论（theory of NP completeness） 研究 **NP 完全问题**的理论。一个问题被确认为具有 NP 完全性，或被称为是 NP 完全问题，意味着该问题是 NP 问题类中“最困难”的问题，是一个固有难解问题。对于这种问题，寻求一个现实可计算的（即多项式时间的）算法，是十分困难的，甚至可能是根本不存在的。历史上第一个 NP 完全问题是可满足性问题，它是由 S. A. Cook 于 1971 年提出的。下面是一个常被人们提起的 NP 完全问题。

旅行商问题 也称货郎担问题。假定一个销售员要到 n 个城市去推销产品。已知各城市间路程 $d_{i,j}(1 \leqslant i,j \leqslant n)$ 和一个界限 B，问是否有一条旅行路线，恰好到每个城市一次，最后返回出发城市，且总路程不超过 B？

旅行商问题还有另一种提法：试求出总路程最短的旅行路线。前者称为判定形式问题，后者称为最优形式问题。可以证明两种形式问题在下述意义下是等价的：如果对于最优形式问题有多项式时间算法，那么也容易找到对于判定形式问题的多项式时间算法，反之亦然。同时还容易看出，问题可以看成由无数个实例组成，一组 n，B 和 $d_{i,j}$，就代表了问题的一个实例。算法必须对问题的一切实例都能给出解答。

对于旅行商问题，从一个城市出发，有 $n-1$ 个城市可作为第二站，选定一个城市后，又有 $n-2$ 个城市可作为第三站，……。这样，总共将有 $(n-1)!$ 条不同的旅行路线。检查每一条旅行路线的总路程，就可以得到问题的解答。但这是一个指数时间复杂度的算法。指数时间算法被公认为不是现实可计算的算法。实际上，对一个底数为 2 的指数时间算法，当 n 为 50 时，即使采用每秒百万次运算的计算机，要计算出解答来，也需要 35.7 年；而当 n 为 60 时，则需要 366 个世纪，这是人们绝对无法容忍的。

一般 NP 完全问题，情况都与旅行商问题相似，都有一个比较明显的指数时间算法。多项式时间算法被公认为现实可计算的算法，那么 NP 完全问题是否有多项式时间算法？这是 NP 完全性理论中的核心问题。

P 一类问题的集合。对类中任一问题，都存在一个确定型图灵机 M 和一个多项式 p，对于该问题的任何（编码）长度为 n 的实例，M 都能在 $p(n)$ 步内，给出对这个实例的回答（参见 **P 类问题**）。

NP 一类问题的集合。对类中任一问题，都存在一个非确定型图灵机 M 和一个多项式 p，对于该问题的任何长度为 n 的实例，M 都能在 $p(n)$ 步内，给出对这个实例的回答（参见 **NP 类问题**）。

多项式时间多一归约 假定给了两个问题 q 和 q_0，如果存在一个确定型图灵机 M 和一个多项式 p，对于问题 q 中任意一个长度为 n 的实例 x，M 都能在 $p(n)$ 步内，计算出问题 q_0 的一个实例 y，使得 x 是 q 中有肯定回答的实例，当且仅当 y 是 q_0 中有肯定回答的实例，就称问题 q 多项式时间多一归约到问题 q_0。

NP 完全问题 如果 q 是 NP 类问题，且 NP 中任意一个问题，都可以多项式时间多一归约到 q，就称 q

为 NP 完全问题，或称q具有 NP 完全性。在现实生活中已发现大量 NP 完全问题，它们分布在计算机科学、数学、逻辑学和运筹学等诸多学科领域中。其中有代表性的，除旅行商问题和可满足性问题外，比较典型的还有划分问题、带优先次序的处理机调度问题、顶点覆盖问题和三维匹配问题等。NP 完全问题总数已达数千个（参见 **NP 完全问题**）。

NP 完全性与 NP = ? P 问题　由于确定型图灵机是非确定型图灵机的特殊情况，因而 P 类中任何一个问题都是 NP 类中的问题。有趣而重要的问题是其反问题：NP 类中任何一个问题是否都是 P 类中的问题？如果是，则 NP = P，否则 NP ≠ P。这就是著名的 NP 是否等于 P 的问题。它是计算机科学中一个非常重要而又历经 20 多年始终未获解决的问题。它的解决将导致一系列理论问题的解决，还将关系到相当一批实际问题是否是“现实可计算的”这样一个大问题。

NP 完全性研究与 NP = ? P 问题有密切关系。已经证明，只要任何一个 NP 完全问题属于 P 类，那么一切 NP 类问题都将属于 P 类，于是 NP = P。实际上，假定q_0是 NP 完全问题且 q_0 属于 P 类。现在考虑任意一个 NP 类问题 q。由于 q_0的 NP 完全性，对于 q 的任给实例 x，在多项式时间内可计算出q_0的一个实例 y，而 x 是否有肯定回答取决于 y 是否有肯定回答。由于 q_0属于 P 类，于是可在多项式时间内判定 y 是否有肯定回答，因而可以推出，在某个多项式时间内，可给出 x 是否有肯定回答的结论。如果要否证 NP = P，只需找出一个属于 NP 类但不属于 P 类的问题。它理应是 NP 类中最难的，即复杂度最高的问题。由于 NP 中一切问题都可以多项式时间多一归约到任一个 NP 完全问题，可以认为 NP 完全问题体现了 NP 类中各种问题计算中的困难性。因此，如果 NP 类中真有不属于 P 类的问题，NP 完全问题应该是首选对象。可以说，如果证明了任何一个 NP 完全问题不属于 P 类，则有 NP ≠ P。遗憾的是，迄今，还未能找到任何一个 NP 完全问题具有多项式时间算法，同时也未能证明任何一个 NP 完全问题不可能有多项式时间算法。即是说，利用 NP 完全问题，既没有证明 NP = P，也没有证明 NP ≠ P。

NP 完全问题的搜寻和证实　NP 完全性的相当一部分研究工作是去搜寻现实世界中可能有的 NP 完全问题，并证明它们的 NP 完全性。当一个问题长久找不到多项式时间算法时，就要认真考虑它是否是 NP 完全问题。很多 NP 完全问题正是这样得到的。证明一个问题是 NP 完全问题的主要途径是证明某个已知的 NP 完全问题可以多项式时间多一归约到这个问题，且该问题属于 NP 类。具体证明方法是多种多样的，常用有限制法、局部替换法以及分量设计法等。

NP 完全问题的求解　对于一个 NP 完全问题，找到多项式时间算法，就相当于证明了 NP = P，而 NP = ? P 问题历时 20 多年经许多人研究，至今仍未解决，这表明对 NP 完全问题寻找多项式时间（即现实可计算的）算法相当困难。另一方面，如果 NP ≠ P，则表明 NP 完全问题根本不可能有多项式时间算法。然而，面对现实生活中大量的 NP 完全问题，人们必须找到有效的解决方法，找到可以应用的算法。许多 NP 完全问题的明显算法是指数时间的穷举搜索法。

解决 NP 完全问题的一类方法就是采用各种方法，尽可能减少这种搜索量。例如采用“分支限界法”或“隐枚举法”，它不能从根本上降低算法的时间复杂度，但可以节约大量时间。此外，还有动态规划法、割平面法和拉格朗日法等。

第二类方法是处理最优化形式的 NP 完全问题。对于这类问题，比较现实的态度是降低最优化要求，求近似最优解，以期得到一个多项式时间算法，一个在容许时间内能得到容许精度的近似最优解的算法。这类方法常常是针对某类问题，甚至是某个具体问题的。它们是一种探索式算法，融会了设计人员的经验和智慧。在这些方法的研究中，对所得解和最优解近似程度的精确分析，以及如果 NP ≠ P 时各种 NP 完全问题的多项式时间算法在解的近似度方面的局限性的探讨，都是重要的研究课题。

与 NP 完全性研究有关的理论课题，除 NP = ? P 问题外，还有 NP 结构的研究、多项式谱系的研究、多项式空间完全性的研究，以及对数空间复杂性研究等。

参考文献

1. Garey M R, Johnson D S. 计算机和难解性：NP 完全性理论导引. 张立昂，沈泓，毕源章，译. 北京：科学出版社. 1987

2. Hopcroft J E, Ullman J D. 自动机理论、语言和计算导引. 徐美瑞，译. 北京：科学出版社，1986

（徐美瑞）

OBJ yuyan

OBJ 语言（OBJ language）　用于书写和测试

代数规约的可执行的形式语言。它采用**抽象数据类型**的始代数的方法刻画其**语义**,用重写规则系统实现。OBJ 由美国加州大学洛杉矶分校 J. A. Goguen 等人于 1977 年提出。

用 OBJ 书写的代数规约由若干对象组成,对象以 OBJ 开始,以 JBO 结束。每个对象由名、类别、操作、变量、等式五部分组成。名是标识该对象的唯一标识符。类别部分给出该对象引入及使用的类别,一个对象可以没有引入的类别。操作部分给出该对象定义的操作符的型构,即其参量类别和结果类别,等式部分刻画了这些操作符满足的性质。变量部分定义了等式中使用的变量的类别。

OBJ 除提供正常操作符外,还提供了异常操作符和恢复操作符,这三类操作符分别在 OK - OPS, ERR - OPS, FIX - OPS 中描述,共同构成操作符部分。异常操作符用于检测出错情况和形成出错信息,恢复操作符用于将异常出错值转换为正常值。

等式部分由 OK - EQNS, ERR - EQNS 和 EQNS 三类等式组成。OK - EQNS 刻画结果为正常值的等式, ERR - EQNS 刻画产生异常值(异常操作符)条件的等式, EQNS 中等式的结果可为正常值也可为异常值,用于刻画恢复操作符的性质。OBJ 解释系统中把等式看成自左到右的重写规则。

OBJ 允许用户定义操作符的语法,进行类别强制转换,允许操作符的重载。在对象的描述中,可以定义隐蔽操作符,它只在对象内可见,而对外不可见。此外,还可指定操作符满足结合律、交换律。对出错情况的处理是 OBJ 的独到之处,通过提供异常操作符,可以比较精确地刻画出错条件及对错误进行定位。

在 OBJ 的基础上, J. A. Goguen 等人于 1987 年提出了 OBJ2。在 OBJ2 中,对一个已有的抽象数据类型,可定义新的操作和公理,对其模型进行扩充。此外, OBJ2 还提供了模块化机制。它们已用于书写机场调度的数据库系统等小规模的实例。

参考文献

Goguen J A, Tardo J J. An introduction to OBJ: a language for writing and testing formal algebraic program specifications. In: Gehani N, McGettrick A. Software specification techniques. Addison - Wesley Publishing Company, 1986

(费宗铭)

Occam yuyan

Occam 语言(Occam language) 一种语句级并行的并行程序设计语言,用于描述多处理器互连网络上的并行算法及其实现。Occam 的主要概念是并行和通信,其主要思想源于 D May 的实验性程序设计语言(EPL)和 C A R Hoare 的通信顺序进程(CSP)。

Occam 没有类型概念,因而也没有变量说明。它的每个语句表示一个**进程**。简单语句描述原始进程;再通过进程构造组成复合进程,因此,程序也是一个进程。并行进程构造使多个进程可并行或并发执行。并行进程间的通信通过通道(channel)进行。输出进程

```
channel! expression
```

将表达式 expression 的值送到通道 channel 上;而输入进程

```
channel? variable
```

从通道 channel 上接收一个值并存入变量 variable。输入输出活动是配对的,输出进程的活动将被推迟直到对应的输入进程开始活动,反之亦然。

除了与常规的语句构造类似的顺序(SEQ)、条件(IF)和循环(WHILE)进程构造外, Occam 还有并行(PAR)和选择(ALT)进程构造。例如,进程

```
PAR
channel? expression
channel! variable
```

表示并行地向通道 channel 上输出表达式 expression 的值,并从该通道上取出存入变量 variable,它等价于赋值语句 variable: = expression,但可用于两个处理机之间的通信。选择类似于条件,但可依赖于其他进程是否正在进行输出。选择中的每一项由一个阀门后接一个进程。阀门由一个可选的真值表达式后接一个输入、延迟进程或空进程(SKIP)构成。例如

```
ALT
red. selected&red. channel? x
out. channel! x
green. selected&green. channel? x
out. channel! x
NOT(red. selected OR green. selected) &SKIP
out. channel! defanlt. value
```

表示当 red 被选中并且 red 通道上有输出,则接收并送到输出通道上;当 green 被选中并且 green 通道上有输出,则接收并送到输出通道上;否则,向输出通道上送默认值。

为了表达时间概念, Occam 引入了局部时钟。

时钟进程 TIME? variable 表示取当前时间,延迟进程 TIME? AFTER expression 表示进程挂起,直到时间超过表达式 expression 的值为止。

Occam 原来是作为大规模并行的 transputer 系统的“**汇编语言**”而设计的。它以其清晰、简洁的结构和同步机制代表了一种并行程序设计的风格。后续的研究包括 Occam 2, Occam 2.1 与 Occam-π。

参考文献

Jones C. Programming in OCCAM. Printice Hall, 1987

（伊波）

OpenGL tuxing biaozhun

OpenGL 图形标准(OpenGL Graphics Standard)　以美国 SGI 公司的 GL 三维图形库为基础的一个通用共享的开放式三维图形工业标准。作为一个优秀的三维图形标准,OpenGL 提供了丰富的绘图命令,利用这些命令能够开发出高性能、交互式的三维图形应用软件。

目前,Microsoft,SGI,IBM,DEC,SUN,HP 等大公司都采用了 OpenGL 作为三维图形标准,许多软件厂商也纷纷以 OpenGL 为基础开发自己的产品,其中比较著名的产品包括动画制作软件 Soft Image 和 3D Studio MAX、仿真软件 Open Inventor、虚拟现实软件 World Tool Kit、CAD/CAM 软件 ProEngineer、地理信息系统软件 ARC/INFO 等。

OpenGL 实际上是一个开放的三维图形软件包,它独立于窗口系统和操作系统,以它为基础开发的应用程序可以相当方便地在各种平台间移植。OpenGL 具有如下 7 种功能:

(1) 建模　OpenGL 除了提供基本的点、线、多边形的绘制函数外,还提供了绘制复杂三维形体(球体、锥体、多面体等)及复杂**曲线**和**曲面**(例如 Bézier、NURBS 等曲线或曲面)的功能。

(2) 变换　OpenGL 的变换功能包括基本变换和投影变换。基本变换有平移、旋转、变比、镜像四种变换,投影变换有平行投影(又称正射投影)和透视投影两种变换。

(3) 颜色模型设置　OpenGL 颜色模型有两种,即 RGBA(A 表示 alpha 颜色分量)模型和颜色索引模型。

(4) 光照和材质设置　OpenGL 有辐射光、环境光、漫反射光和镜面光。材质是用光反射率表示的。场景中物体最终反映到人眼的颜色是光的红绿蓝分量与材质红绿蓝分量的反射率相乘后形成的颜色。

(5) 纹理映射　利用 OpenGL 纹理映射功能可以相当逼真地表达物体表面细节。

(6) 位图显示和图像增强图像功能　除了基本的复制和像素读写外,还提供融合、反走样和雾的特殊效果处理。以上 3 条可使被仿真物更具真实感,增强图形显示的效果。

(7) 双缓存动画　双缓存即前台缓存和后台缓存,后台缓存用来计算场景、生成画面,前台缓存显示后台缓存已生成的画面。此外,利用 OpenGL 还能进行深度提示、运动模糊等特殊效果的处理,并实现消隐算法。

1992 年 1 月 SGI 公司的 Mark Segal 和 Kurt Akeley 发布了 OpenGL 1.0。2004 年 9 月发布的 OpenGL 2.0 增加了对高级着色语言 GLSL 的支持,从而使得开发者可以更容易地利用可编程图形硬件的各项特性。2006 年 Khronos 小组接手了 OpenGL 的开发。2008 年 7 月发布的 OpenGL 3.0 增加了对帧缓存对象、顶点数组对象、纹理数组和基于硬件的实例化的支持。2010 年 3 月发布了 OpenGL 4.0,加入了两个新的着色步骤并提供了对可编程三角化的支持,同时还在着色器中支持了 64 位双精度浮点运算。

参考文献

1. Shreiner,等. OpenGL 编程指南(原书第 7 版). 李军,等译. 北京:机械工业出版社,2010

2. 中国游戏开发者. http://mays.soage.com/develop/opengl/200112/GLInduction.htm

（张明敏　任重）

OSI anquan tixi jiegou

OSI 安全体系结构(OSI security architecture)　OSI 安全体系结构的研究始于 1982 年,并于 1988 年完成,其成果表现在 ISO 7498-2 标准中,作为 OSI 基本参考模型的补充。OSI 安全体系结构中定义了为实现安全通信与访问提供的 5 种可选的安全服务(security services)、8 种特定的安全机制(security mechanisms)和 5 种普遍性的安全机制,并且定义了安全服务与安全机制的关系以及这些安全服务在 OSI 七层模型中的配置。OSI 安全体系结构还规定了安全管理的基本概念与内容。按照 OSI 安全体系结构的基本思路,网络的安全需求是通过使用分布在各层的安全服务来满足的。同时,由于这些安全服务从功能分解的角度又分别有一些公共的部分,因此将这些分解的功能定义为安全机制,它们

分别可以支持一个或几个安全服务。至于安全机制的实现则像 OSI 协议的实现一样,可以有多样性,且可随技术的进步而进化。

根据 ISO 7498-2,OSI 安全体系结构定义的可选安全服务分别是:鉴别、访问控制、数据保密、数据完整性和抗否认。

鉴别服务包括对等实体鉴别(含身份认证)和数据源鉴别。前者包含单向鉴别和双向鉴别,要求能够准确无歧义地将通信一方或双方辨别出来,证实其真实性;后者则要求能够证实数据来源的真实性。

访问控制服务提供授权控制,能够定义并实施对不同用户访问网络与信息资源的权限控制,防止对资源的未授权使用。

数据保密服务提供对网络中存储或传输的数据进行加密保护,具体包括连接机密性,提供整个连接上所有数据的保护;无连接机密性,提供单个数据的保护;选择字段机密性,提供部分数据内容的保护;通信业务流机密性,提供通信活动保护。

数据完整性服务可以防止通过网上传输的数据被修改、删除、插入、替换或重发,以保证合法用户接收和使用该数据的真实性。具体的要求包括带恢复连接完整性,向整个连接上的所有数据提供完整性保护,发现完整性错误时可进行恢复;无恢复连接完整性;选择字段连接完整性,在整个连接上对部分数据内容提供完整性保护;无连接完整性,对单个数据提供完整性保护;选择字段无连接完整性,对单个数据的部分内容提供完整性保护。

抗否认服务可以使防止参与通信的各方对通信动作和数据源的否认,具体包括有源点抗否认,可向第三方证明数据确实是由该发送者发出的;交付抗否认,可向第三方证明数据确实已经交付给接收者。

ISO 7498-2 为 OSI 安全体系结构定义了加密机制、数字签名机制、访问控制机制、数据完整性机制、鉴别交换机制、通信业务填充机制、路由控制机制和公证机制等 8 种特定的安全机制,以及可信功能度、安全标记、事件检测、安全审计跟踪、安全恢复等 5 种普遍性的安全机制。

加密机制实现数据加密,能为数据提供机密性保护,涉及的主要技术内容为加密算法和密钥管理。加密机制既单独作为一种机制运行,也作为后面其他机制的一部分,起到支撑的作用。

数字签名机制用于对特定数据单元生成可验证的数字签名,以构成该数据的完整性和真实性的证据。

访问控制机制用于决定和实施一个已鉴别身份的实体的访问权。该机制包括基于规则策略的访问控制方式和基于身份标识的访问控制方式两种。前者除了实体的身份标识,还可以依据实体可提供的其他相关属性与被访问对象的各种相关属性的匹配情况,更适合于像因特网这样的开放访问环境;而后者则是传统的“用户名 + 口令”的访问控制方式。

数据完整性机制是数字签名机制的一种应用,用于保护数据不被篡改和不当使用(例如被回放-用一个过去的数据来冒充当前的数据)。数据完整性分两个方面考虑,单个数据或其中字段的完整性以及成组数据或其中字段的完整性。例如前者考虑一个报文的内容的完整性,而后者要考虑一组报文中报文之间的错序和报文的缺失。

鉴别交换机制用于支持实体鉴别,而源鉴别通常依靠数据完整性机制。鉴别交换机制在加密机制的支持下,通过在实体间交换鉴别信息来达到单向鉴别或双向鉴别的目的。

通信业务填充机制通过往信道中注入伪装流量来隐藏正常的通信活动和通信内容,用于抵御通信业务分析攻击。它只有在通信业务受到数据保密服务保护时才有效。

路由控制机制用于保护网络路由,使得在遭到攻击时仍然能够维持工作。该机制要求路由能够动态地或预定地选取,以便选用物理上安全的子网络、中继节点或链路。在检测到持续的攻击时,端系统可以指示网络服务提供者改变路由;也可以根据数据的安全标记要求其通过或回避网络中的某些路由。

公证机制用于支持抗否认服务,在多个相关实体之间提供特定的数据信息交换过程,形成第三方可验证的证明信息,可用于数据交换过程、数据内容和数据源的纠纷仲裁。

以上 8 项特定的安全机制是为了特定的安全服务所设置的。除此之外,还有以下 5 种普遍性的安全机制,虽然不是为特定的安全服务而设置,但是与系统所要求的安全级别直接有关,一些机制与安全管理有关。

可信功能度机制可以扩展其他安全机制的范围,或建立这些安全机制的有效性。任何功能,只要它是直接提供安全机制或提供对安全机制的访问,都应该可以用可信功能度来刻画。

安全标记机制为资源(包括数据)提供与安全

有关的标记功能,例如为访问控制机制提供资源敏感性标记,可以作为访问控制策略定义的依据。

事件检测机制是一种管理功能,与安全有关的事件检测包括检测对安全的明显侵害(如特定的安全攻击、特别选择的事件以及事件发生计数的溢出等)的检测,也可能包括对普通“正常”事件的检测(例如一次成功的访问)。

安全审计跟踪机制对系统的记录与行为进行独立的评审考查,经过事后的审计来检测是否存在违反了安全策略的安全事件,评估相应的损失,调查安全事件发生的原因并以此发现系统中存在的安全漏洞。

安全恢复机制处理来自于诸如事件处理与管理功能等机制的请求,并把恢复动作作为使用一组规则之后的结果。该机制试图将系统的恢复功能规范化,提出将恢复动作分为三种:立即动作,可造成当前操作的中断,例如断开连接;暂时动作,造成一个实体暂时失效,例如设置防火墙的过滤规则以封堵某个 IP 地址的流量;长期动作,发生永久性的变化,例如改变密钥。

OSI 的安全管理涉及与 OSI 有关的安全管理和 OSI 管理的安全两个方面,面向从属于统一的安全策略,受单一授权机构管理的安全域。OSI 安全管理活动有三类:系统安全管理,涉及总的 OSI 环境安全方面的管理(包括安全策略管理),与其他管理功能的交互等。特定安全服务的管理,包括具体安全目标和安全策略的确定、对特定安全机制的使用方式等。安全机制管理,实现对那些特定安全机制的管理。

参考文献

ISO 7498-2—1989. Information processing systems; Open Systems Interconnection; basis reference model; Part 2: Security architecture (龚俭 胡道元)

PASCAL yuyan

PASCAL 语言(PASCAL language) 在 **ALGOL 60 语言**的基础上发展起来的一种重要的**程序设计语言**。PASCAL 是 Philips Automatic Sequence Calculator 的缩略语,它是由 N. Wirth 设计的,1971 年正式发表。

1965 年 IFIP 工作组提出关于 ALGOL60 后继语言的讨论。IFIP2.1 小组承接了提出其后继语言的任务,不久即分为两派。一派雄心勃勃想要在语言设计上另树丰碑,代表人物是 van Wijngaarden,由他设计了 ALGOL 68。而另一派认为 ALGOL 68 太复杂不易推广。N. Wirth 设计的 PASCAL 语言关心的是概念级、用户级和实现级的简明性。它已广泛地用于大学程序设计语言的教学和许多系统软件及某些应用软件的开发中。

其主要特点是:

(1) 丰富的数据结构和构造数据结构的方法。除了整型、实型、布尔型和数组外,还提供了字符、枚举、子域、记录、集合、文件、指针等类型。由这些数据结构可以方便地描述各种事务元。

(2) 简明灵活的控制结构。具体的结构语句有复合语句、如果语句、情况语句、While 语句、Repeat 语句、For 语句和处理记录变量的分量的缩写形式——With 语句。

(3) 它可称为第一个结构化程序设计语言。

(4) 编译运行效率高。

(5) 有利于书写程序设计语言的编译程序。

在 PASCAL 中没有引入模块的概念,不利于开发大型软件,这是它的不足之处。

BSI(英国标准化学会)1982 年正式出版英国标准 BS 6192(即 IS 07185),全名为“Specification for Computer Programming Language PASCAL”。我国 1987 年正式生效国家标准(编号为 GB 7591—87),全名为《程序设计语言 PASCAL》。1993 年公布修订标准 ISO/IEC 7185: 1990 文本。

参考文献

1. 郑国梁,钱士钧. PASCAL 语言——标准丛书. 北京:中国铁道出版社,1989

2. Wirth N. The programming language PASCAL. New York: Springer-Verlag,1971 (郑国梁)

PDL yuyan

PDL 语言(PDL language) 一种设计性语言。它是由美国的 S. Caine 和 K. Gordon 在 1975 年提出的。PDL 是 program design language(设计性程序语言)的缩写,用于书写软件设计规约。它是软件设计中广泛使用的语言之一。

用 PDL 书写的文档是不可执行的,主要供开发人员使用。PDL 描述的总体结构和一般的程序很相似,包括数据说明部分和过程部分,也可以带有注释等成分。但它是一种非形式的语言,对于控制结构的描述是确定的,而控制结构内部的描述语法不确定,可以根据不同的应用领域和不同的设计层次灵活选用描述方式,也可以用自然语言。

PDL 语言书写的模块结构如下：

```
PROCEDURE〈过程名〉(〈参数表〉)
〈数据说明部分〉
〈语句部分〉
END〈过程名〉
```

数据说明部分形式为：

```
DECLARE〈数据说明表〉
```

数据说明表由一串说明项构成，每个说明项形如：

```
〈数据项名〉AS
〈类型字或用户定义的类型名〉
```

语句部分可以包括：赋值语句、if - then - else 语句，do - while 语句、for 语句、case 语句、调用语句、返回语句等。与一般程序模块不同，其语句中除描述控制结构的关键字外，书写格式没有严格定义。自然语言书写的注释可以插在任意位置。

下面是一个 PDL 描述的结构的示例：

```
    PROCEDURE
    EXAMPLE
    (arg 1,arg 2,...)
    DECLARE   arg 1     AS
    CHAR ARG
              arg 2     AS
SCALAR ARG
              ...
    〈初始化操作〉
    DO WHILE〈条件 1〉
        IF〈条件 2〉
          THEN BEGIN
              〈具体操作〉
                END
          ELSE〈具体操作〉
    ENDIF
    〈具体操作〉
    ENDDO
    RETURN
END EXAMPLE
```

PDL 描述接近自然语言（英语），易理解。它虽然不如图形化的设计描述直观，但和可执行的程序具有类似的结构，因此，便于实现借助计算机自动转换为可执行的程序代码，已经研制出针对特定语言的自动工具。

参考文献

Caine S, Gordon K. PDL — a tool for software design. In: Proc. National Computer Conference, AFIPS Press, 1975, 271-276

（陈道蓄）

Perl yuyan

Perl 语言（Perl language）　一种高级通用的、解释执行的动态程序语言。

Perl 语言具备强大的正则表达式处理能力，适用于系统管理、网络应用、软件工程等诸多领域。Perl 由 Larry Wall 等人设计，迄今已发布了 5 个主要版本（Perl1 到 Perl5），目前持续更新的仅为 Perl5 以及尚处于开发阶段的 Perl6（已于 2010 年 7 月发布了实验版本）。

Perl 遵循一题多解、简单问题简单处理、复杂问题能够处理的设计原则。它的语法和控制结构基本借鉴于 **C 语言**；链表、哈希表和正则表达式等复杂结构则分别源自 Lisp、awk 和 sed；重写 Perl5 时又增加了对模块和对象的支持。由于融合了 C、sed、awk、UNIX shell 等多种语言的特征，Perl 语言易于入门，编程风格灵活。Perl 采用动态类型检查机制，程序员无须事先声明变量的类型。下述代码即为一个完整的 Perl 程序，可以在标准输出设备上打印“Hello World!”

```
print "Hello World! \n";
```

Perl 用 $ 导引来标识标量，用 sub 语句说明子过程，用 package 语句建立一个新的命名空间并构成包。Perl 提供模块概念支持包的复用。复用时，需要使用 use 语句。Perl 程序结构分为以下三类：正常语序的顺序结构，由 if、else、elsif 组成的条件分支结构和由 while、until、do-while、for 和 foreach 导引的循环结构。Perl 还为表达式提供了简洁高效的内置匹配模式，如匹配、替换、转换等，简化对数据和正文的处理。

Perl 语言的设计初衷是实现对正文的高效处理。至今，Perl 已发展成一种复杂通用的**程序设计语言**。1995 年，Perl 社区引入了程序库 CPAN。目前，CPAN 汇集了 Perl 源码、使用文档以及九千多个由第三方开发的 Perl 模块供用户使用。2000 年，Larry Wall 提出开发 Perl6 计划。Perl6 以 Parrot 为底层平台，在设计之初就考虑了对 unicode 和多线程的支持，目标是更加清晰、高效且易扩展。

参考文献

1. Wall L, Christiansen T, Orwant J. Schwartz: Programming Perl. 3rd ed. O'Reilly Media, 2000

2. Schwartz R L, Phoenix T, brian d foy. Learn-

ing Perl. 5th ed. O'Reilly Media, 2008

（李丰　霍玮）

PHP chaowenben yuchuli chengxu

PHP 超文本预处理程序（PHP hypertext preprocessor）　一种跨平台多用途的**脚本语言**。

PHP 的发展始于 1994 年，程序员 Rasmus Lerdorf 开发了一个 perl 脚本集，并称为"个人主页工具"，并在 1995 年以 personal home page tools（PHP Tools）开始对外发表第一个版本，形成了 PHP1.0。在后续发展中逐渐丰富了语言成分，并重写了解释程序。目前 PHP5 是唯一稳定的开发版本。

PHP 最主要的应用是动态网页实现以及网络应用开发，也可以作为命令行脚本处理文本或者用于桌面应用开发。PHP 具有丰富的功能，高效的执行和广泛的应用。PHP 能够实现所有的 **CGI** 或者 **JavaScript** 的功能，而且支持几乎所有流行的**数据库**以及**操作系统**。根据 2007 年 4 月的统计数据，PHP 已经被安装在超过 2000 万个网站和 100 万台服务器上。现有的 PHP 标准由 PHP 工作组和开放源代码社区维护，并使用 PHP License 作为许可协议。

PHP 语言的风格主体类似于 **C 语言**，结合了 C、Java、Perl 以及 PHP 自身的新语法。以下是 PHP 的 hello world 程序。

```
<?php
    echo 'Hello World!';
?>
```

PHP 有 4 种标量类型：整型、浮点型、布尔型、字符串型；2 种复合类型：数组、对象以及 2 种特殊类型：NULL、资源。PHP 中的变量以"＄"后接变量名称来表示，变量名称区分大小写。而且，PHP 中的变量不需要说明类型，PHP 根据变量的值自动识别变量的类型。PHP 的面向对象功能目前比较完善。PHP 中还提供了丰富的内置函数，实现了文件处理、FTP、字符串处理等多种功能。此外，PHP 还提供了众多扩展库函数，如各种数据库链接函数、数据压缩函数、图形处理函数等。PHP 程序由一系列语句构成，语句有赋值语句、函数调用、循环以及条件语句（语句内容可以为空）等，语句通常以分号结束。PHP 解释程序只分析从"<? php"到"? >"之间的代码，不包含在此间的内容则会直接提交。此外，还可以用"{"和"}"将一组语句封装成一个语句组。它具有以下三类程序结构：正常语序的顺序结构，由 if、else、elseif、switch 以及嵌套 if 组成的分支结构和由 while、for、do... while 导引的循环结构。

总体来说，跨平台以及数据库的支持、与主流语言语法风格的兼容、高效的执行机制、丰富的开发库支持成为 PHP 广为流行的主要原因。

参考文献

PHP Manual. Retrieved, 2011-5-20

（陈聪明　霍玮）

P lei wenti

P 类问题（class P of problems）　多项式时间内可解决的问题类，它是**计算复杂性理论**中十分重要的问题类。

在计算复杂性理论中，问题由无数个实例组成。考虑分割问题：给定 n 个自然数，是否可将它们分成两部分，使两部分自然数的和数彼此相等。不同的 n，不同的自然数，就构成了不同的实例。

每个实例经过编码，可表成一个字符串。例如，将 n 个自然数表成二进制数，排成一行，中间用"#"号隔开，分割问题的每个实例，就表成字母表 $\sum = \{0,1,\#\}$ 上的一个字符串。在这些字符串所代表的实例中，有些可以分成和数相等的两部分。这些字符串就构成一个语言（参见**形式语言**）。

一个语言 L 称为属于语言复杂性类 P，如果存在一个确定型**图灵机** M 和一个多项式 p，M 在 $p(n)$ 时间内接受 L，亦即对于每个长度为 n 的字符串 w，只要 w 在 L 中，M 就能在 $p(n)$ 步内停机并接受 w。

如果一个问题所对应的语言属于语言复杂性类 P，就称它是 P 类问题。其直觉意义是：有一个算法和一个多项式 p，若实例的编码长度为 n，则该算法可在 $p(n)$ 步内给出问题的解答。正整数是否可被 4 整除的问题，是一个简单的 P 类问题。因为只要将该数表示成二进制数，图灵机扫视一遍这个输入，以确定最低两位数是否为 0，即可给出问题的答案。这个算法只需线性时间。现实中有大量 P 类问题。大多数数值计算问题都是 P 类问题。其他如排序问题，任务长度相等的多处理机调度问题，含删除、插入、替换操作的字符串到字符串的修正问题，箱容量固定的装箱问题和二维匹配问题等。

P 类问题公认为是计算机上现实可解决的问题。在表 1 中，假定每秒进行一百万次运算。容易看出，对于多项式时间复杂性函数，即使方次达到 5，当 n 增大时，计算时间平稳增长，且是人们可以接

受的。而对于指数时间复杂性函数,即使底为 2,计算时间也随 n 急剧增长,常常达到人们无法容忍的地步。因此,人们公认,多项式时间可解问题是现实可解的问题。

表 1 多项式时间与指数时间复杂性函数的比较

复杂性函数	问题规模 n			
	10	30	50	60
n	0.01 ms	0.03 ms	0.05 ms	0.06 ms
n^3	1 ms	27 ms	125 ms	216 ms
n^5	100 ms	24.3 s	5.2 min	13.0 min
2^n	1 ms	17.9 min	35.7 年	366 世纪

参考文献

Garey M R, Johnson D S. Computers and intractability. San Francisco, CA: Freeman, 1979

(徐美瑞)

PROLOG yuyan

PROLOG 语言(PROLOG language) 一种***顺序逻辑程序设计语言***。PROLOG 是 programming in logic 的缩略语。PROLOG 程序有两类成分:纯逻辑成分与非逻辑成分。纯逻辑成分由有限多个 ***Horn 子句***组成。Horn 子句是最多有一个正文字(原子公式,如 P(X))的一阶子句。单个正文字的子句称为事实;有一个正文字和若干个负文字(原子公式的否定,如 ~P(X))的子句称为规则;仅有若干负文字的子句称为问题。纯逻辑成分的计算成分的计算和控制由 PROLOG 系统实现,主要是合一、归结和回溯。

PROLOG 中非逻辑成分提供许多辅助功能控制特性,例如:cut 操作用于控制,允许用户干预搜索进程;输入输出功能以及类似其他高级程序设计语言中的过程调用功能等。PROLOG 总是按书写的次序搜索规则操作符。“,”表示顺序执行;“is”表达式中所有变量必须赋值才能开始计值。

第一个 PROLOG 解释程序于 1972 年由法国马赛大学 A. Colmerauer 和 P. Roussel 小组实现。1977 年英国爱丁堡大学 D. H. D. Warren 实现了 PROLOG 编译程序,执行效率较高,达到实用水平。

参考文献

Clocksin W F, Mellish C S. Programming in Prolog. Springer-Verlag, 1981

(胡运发)

PSL yuyan

PSL 语言(PSL language) 一种问题陈述语言。PSL 可以按照一定的结构来描述用户对软件系统的功能需求和性能需求。PSL 及其相应的支撑系统 PSA 最早出现在 1971 年,其第一个版本主要使用在美国密歇根大学的 IS DOS 项目之中,其主要目的是用于信息系统的需求定义与分析。它是需求定义语言及其机器支撑方面的早期工作之一,并曾得到较广泛的使用。

PSL 从实体及其相互间关系的角度来刻画系统的输入输出、系统结构、数据结构、数据流程、系统规模、动态行为、系统性质、项目管理等各个方面。具体说来,在数据方面,PSL 提供了 ENTTTY、CONSISTS OF、CONTAINED IN、DERIVED BY、UPDATED BY 等实体或关系来描述数据对象的名、数据结构和数据流程。在数据处理方面,PSL 提供了 GENERATES、RECEIVES、PROCEDURE、DERIVES、UPDATES 和 USES 等实体或关系来刻画处理的名、系统的输入输出、数据流程、处理方式和动态行为等。

严格说来,PSL 是一种半形式化的语言,它在某些局部方面允许用户使用自然语言。例如,在描述处理时,可在 PROCEDURE 中用自然语言描述控制结构方面的信息。在描述数据时,可在 DESCRIPTION 中采用非形式化的方式描述数据的各个方面。

PSA 程序是一种问题陈述分析程序。PSA 是 problem statement analyzer 的缩略语。PSA 可对用 PSL 书写的需求进行分析,产生有用的文档。

PSA 是一种基于数据库的、用于支持 PSL 的交互式工具。它保存用 PSL 书写的软件需求,并对之作语法检查和一致性检查,从而形成一个由计算机管理的字典。PSA 可按照用户的需求对相应的字典进行查阅分析,并可产生各类文档,如格式化的问题描述、数据流图等。

参考文献

Teichroew D, Hershey E A. PSL/PSA: a computer-aided technique for structured documentation and analysis of information processing systems. IEEE Trans. on Software Engineering, 1977, SE-3(1)

(吕建 陶先平)

Python yuyan

Python 语言(Python language) 一种强调程序易读性的、解释型通用高级**程序设计语言**,由 Guido van Rossum 于 20 世纪 80 年代末提出并开始实现。Python 强调代码的易读性,把简明、清晰的语法与一组完善且容易理解的标准库相结合,能够轻松完成很多常见的任务。Python 程序采用解释执行方式,通过 Python 虚拟机可运行于各种平台,通常作为脚本语言用于 Web 应用编程(也可作为非脚本语言和第三方提供的工具生成单独可执行的代码)。

Python 支持命令式、面向对象、函数式、面向方面以及类属(泛型),采用动态类型和自动内存管理机制。Python 在编码风格上的一个显著特点是其代码块(block)的缩进语法,即代码块不需要用{、}或 begin、end 来指出,而是通过增加缩进(空格)来表示语句块的开始,减少缩进则表示语句块的结束。由于 Python 程序是解释执行的,因此,影响其执行效率(这不是 Python 设计理念所关心的),当然,程序中需要高效执行的代码可用其他语言(如 C/C ++)来完成(作为可扩充的标准库)。Python 语言的核心只包含数字、字符串、表、字典、文件等常见类型和函数,而 Python 标准库提供了系统管理、网络通信、文本处理、数据库接口、图形系统、XML 处理等额外的功能。Python 社区还提供了大量的第三方模块,可供使用。

Python 的主要基准实现是 CPython,它是一个由社区驱动的自由软件,目前由 Python 软件基金会管理。Python 的主要版本是 2.0(2000 年发布)和 3.0(2008 年发布),由于 3.0 与 2.0 存在较大的不兼容性,其间存在 2.6 和 2.7 两个过渡版本,它们提供了从 2.0 到 3.0 的兼容性过渡方案。

参考文献

1. Wesley J. Chun. Python 核心编程. 2 版. 宋吉广,译. 北京:人民邮电出版社,2008

2. Mark Lutz. Python 学习手册. 4 版. 李军,刘红伟,等译. 北京:机械工业出版社,2011

(陈家骏)

SIMULA 67 yuyan

SIMULA 67 语言(SIMULA 67 language) 第一个面向对象程序设计语言,由挪威计算中心(Norwegian Computing Center)的 Ole-Johan Dahl 和 Kristen Nygaard 设计。

1962 年春,Nygaard 和 Dahl 开始考虑设计一种用于离散事件仿真的程序设计语言。他们先基于 ALGOL 60 语言设计了 SIMULA I,并于 1964 年底实现第一个编译程序原型。在 SIMULA I 中,系统建模为一组并行共存且相互交互的进程(process)。随后 Dahl 和 Nygaard 着手将 SIMULA I 一般化为一个新的通用程序设计语言。其一个关键想法是将 SIMULA I 的进程和 C. A. R. Hoare 的记录类型(record class)统一为类(class),并引入子类的概念。在 1967 年的 IFIP 工作会议上两位作者发表了关于这个新语言的论文,此语言后来被称为 SIMULA 67。

SIMULA 67 具备了后来被称为"面向对象"的多项基本特征。程序中用**类**来定义一组存在于运行时刻的对象的公共的结构和行为。一个类(称为子类)可继承另一个类(称为父类)。子类在概念上是对父类的特化,在功能上是父类的扩展,在实现上可复用父类的代码。SIMULA 67 支持对象方法的动态绑定:对于通过父类类型的对象引用来实施的**方法**(称为 procedure)调用,若该方法说明为 virtual,则并不在编译时刻将其直接确定为该父类定义的相应方法,而是在运行时刻依据该引用所指对象的实际类型来确定实际调用的方法。然而,SIMULA 67 语言缺乏有力的信息隐蔽设施,因而其对封装的支持不完整。类和数据抽象之间的联系亦有待数年后由 Hoare 明确指出。

SIMULA 67 程序由一组类和主程序组成。类可以嵌套,每个类除了属性和方法外还可有执行体,执行体在对象创建时被执行。通过一种协同例程机制,SIMULA 67 对象可以准并行地执行。

SIMULA 67 在仿真领域之外的应用并不广泛,然而它以超前于时代的方式开创了面向对象的程序设计风范,对程序设计和软件工程的研究和实践产生了深远的影响。后来 Smalltalk,C ++,Java 等许多广泛流行的**面向对象语言**均直接或间接继承自 SIMULA 67,而面向对象的方法亦从程序设计扩展到软件生命周期的各个阶段,并成为软件开发的主流方法。Dahl 和 Nygaard 因在面向对象程序设计方面的贡献获 2001 年图灵奖。

参考文献

1. Dahl O-J. The birth of object orientation: the Simula languages. In: Owe, Krogdahl S, Lyche T, eds. From Object-Orientation to Formal Methods: Essays in Memory of Ole-Johan Dahl. LNCS 2635, Springer Verlag, 2004: 15-25

2. Holmevik J R. Compiling SIMULA: a historical study of technological genesis. IEEE Annals in the History of Computing, 1994, 16(4): 25-37

(马晓星)

Smalltalk yuyan

Smalltalk 语言(Smalltalk language) 一种**面向对象语言**。1972 年由美国 Xerox 公司研究中心(PARC)以 A. Kay 为首的一个软件概念小组,在 Flex 系统的基础上研制成功。后经不断的构思、试验和改进,陆续推出若干版本,其中最具影响的是 1981 年推出的 Smalltalk - 80,但直至 1984 年才作为产品公开。

Smalltalk 有 5 个核心概念,即**对象**、**类**、实例、消息和方法。对象是面向对象系统的唯一元素,它的外部特征包括内部使用的若干私有变量和一组方法。类描述了性质相似的一组对象。类的每个对象称为该类的一个实例。消息是发送者(对象)传递给接收者(对象)的请求,要求接收者执行所指操作。方法描述了操作的实现细节。

继承性是 Smalltalk 的特色,它指的是,子类继承父类的一切属性和操作,整个系统的数据是通过子类机制组织成树型结构的。这种机制为信息共享提供了有效的支持。

Smalltalk 的基本语法结构是表达式。表达式是一字符序列,它描述的对象称为表达式的值。Smalltalk 共有 4 种表达式:

(1) 文字表达式　它描述的对象是一个确定的常量,即总是代表同一个对象;

(2) 变量名表达式　它描述的对象是可供使用的变量,变量之值指该变量当前所指的对象;

(3) 消息表达式　它描述传送给接收者的消息,其值由该消息所引用的方法来确定;

(4) 块表达式　它描述的对象表示一系统被延迟的活动,常用来实现各种控制结构。

在 Smalltalk 中,建立程序就是根据类创建对象,执行程序就是不断向对象发送消息的过程。

Smalltalk 的主要特点是: ①信息表示与处理的高度一致性; ②弱类型语言; ③比较完善的抽象机制; ④语言融合于环境之中。

Smalltalk 在继承和并发等方面较弱。然而,它既推动了混合型 OO 语言(如 C + +)的开发,又促进了对纯 OO 语言(如 Eiffel)的深入研究。自 1986 年以来,还推出了很多 Smalltalk 增强版本。例如,运行在微机 DOS 下的 Smalltalk /v,增加并行性的 Concurrent Smalltalk。

参考文献

1. Goldberg A, Robson D. Smalltalk - 80: the language and its implementation. Addison -Wesley, 1983

2. Goldberg A. Smalltalk - 80: the interative programming environment. Addison - Wesley, 1983

3. 徐家福,等. 对象式程序设计语言. 南京: 南京大学出版社, 1992

(郭浩志)

SNOBOL yuyan

SNOBOL 语言(SNOBOL language) 一种**串处理语言**。SNOBOL 是 string-oriented symbolic language 的缩略语。SNOBOL 设计于 1962—1963 年,主要目的在于满足串处理的一般需求,另一重要的考虑是基于对符号化的数学表达式的处理。

在 SNOBOL 中串表示为字符序列。串括以引号,但引号并非串的组成部分。例如:'Ruanjian'。这样的串特定表示为“字面常量”,串也可以用名来表示,如:

```
FIRST = 'Cheng'
```

它将串 Cheng 赋值给名 FIRST。串的字符个数不限,存储管理自动进行,无须进行说明定义。用串的并置表示并置运算,这种串可以用字面常量形式给出或用作某一个名的值,例如:

```
FULLNAME = FIRST 'San'
```

它将串 Cheng San 赋值给名 FULLNAME。为清楚起见,空格用' '表示,也作为一种字符。

一个 SNOBOL 程序由语句序列组成,它有三种语句形式:赋值,模式匹配和替换。形式如下:

```
标号　主项 = 目标　　　goto
标号　主项　模式　　　　goto
标号　主项　模式 = 目标　goto
```

标号标记语句,主项表示语句操作的对象,goto 控制程序执行流,且是任选的。赋值语句是将目标值赋给赋值号左边的变量名;模式匹配语句针对模式是否在主串中出现搜索制定的字符串;替换语句修改模式匹配的部分。

模式匹配是 SNOBOL 语言的一种最重要的基本功能。它是一种搜索主项中是否出现模式所指定的子串的过程。模式匹配是通过模式匹配语句实现的。

例如：　Temp = 'Programming'

那么模式匹配语句

LAB1 Temp 'amm'

从左边开始扫描主项 Temp，搜索是否出现由模式指定的字符串 amm。如果指定的字符串在 Temp 上被找到，则模式匹配成功，反之模式匹配失败。因为在 Temp 中可以找到字符串 amm，上面的模式匹配是成功的。

带置换串的模式匹配语句是模式匹配和赋值相结合，先匹配一个子串，如果模式匹配成功再进行替换。语句

FULLNAME'Cheng' = 'Zhang'

将子串 Cheng 用 Zhang 替换，它将 FULLNAME 的值换为 Zhang San。间接引用表示一个串可以进行运算，然后其值作为名使用。具体形式是以单目运算符 $ 符后一个串的值作为名。例如：

```
X = 'NUM'
      N = '2'
      HOLIDS\AY = XN
       $ HOLIDAY = 'The Spring Festival'
```

首先将 NUM 2 赋值给 HOLIDAY，然后将值 The Spring Festival 赋值给 NUM 2。这种间接引用操作，类似于**汇编语言**中的间接地址，它提供了一种在执行期间构造数据名的方式。

表达式是运算符和操作数的组合。运算符采用基本的算术运算符和串运算符，操作数为整或实数、字符串和变量。

GOTO 域出现在语句的最末位置上，它用来控制程序执行的转移。GOTO 可以是无条件地转向一个具有标号的语句，或是根据模式匹配的成功与失败条件地转向。用模式匹配的条件特征可以编制循环。

在 1965 年，SNOBOL 3 替代 SNOBOL。SNOBOL 3 类似 SNOBOL，但具有一些附加的特征，包括一些嵌入式函数和程序员自定义的递归函数的机制，SNOBOL 3 同样在 1967 年由 SNOBOL 4 替代，SNOBOL 4 成为广泛使用和实用有效的串处理语言。

参考文献

Griswold R E, Poage J F, Polonsky I P. The SNOBOL 4 programming language. Prentice Hall Inc., 1971

（郑国梁）

SSH（secure shell）

SSH（secure shell）　由 IETF 网络工作组制定的一个加密通信协议，用于在不安全的网络环境中为两台互联的网络主机提供一个安全信道，在该信道上可以进行安全数据通信、远程命令行登录、远程命令执行和执行其他安全的网络服务。根据协议的规范不同可以分为两个主版本 SSH-1 和 SSH-2。最常见的 SSH 应用是类 Unix 系统的远程命令行登录和远程命令执行功能，用于代替 Telnet 和其他一些不安全的 Shell 协议如 rsh 和 rexec，这些协议明文发送包括密码在内的敏感信息，容易受到中间人攻击和嗅探攻击。同时 SSH 协议还支持隧道、TCP 端口转发和 X11 连接等功能，并可以通过相关联的 SSH 文件传输（security file transfer，SFTP）协议和安全复制（security copy，SCP）协议进行文件传输。SSH 协议的标准端口是 TCP 协议的端口 22。

SSH 协议包含三个主要部分：

（1）SSH 传输层协议（SSH transport layer protocol，SSH-TRANS）　该部分协议负责服务器认证以及通信的保密性和完整性，并提供压缩传输内容的选项。传输协议一般是运行在 TCP/IP 协议上，也可以运行在其他协议簇上。

（2）SSH 用户认证协议（SSH user authentication protocol，SSH-USERAUTH）　该部分协议负责服务器对客户端用户的认证，它运行在 SSH-TRANS 协议之上。

（3）SSH 连接协议（SSH connection protocol，SSH-CONNECT）　该部分协议通过复用加密的传输信道，为上层应用提供多个逻辑信道，它运行在 SSH-USERAUTH 之上。

在 SSH 协议运行时，客户端首先通过 SSH-TRANS 向服务器端发起请求建立安全的传输协议连接，然后进行用户认证，最后通过连接协议开始应用，连接协议的标准功能包括交互登录会话、远程命令的执行、转发 TCP/IP 连接等功能。

SSH 协议运行时，用户认证方式有如下几种：

（1）基于口令的验证方式　通过输入用户名和密码的方式进行远程机器的登录验证。

（2）基于公共密钥的安全验证方式　通过生成一对公组和私钥来实现用户的登录验证。

（3）基于键盘交互的验证方式　服务器向客户端发送提示信息，然后由客户端根据相应的信息通过手工输入的方式发还给服务器端。

SSH 最初由芬兰人 Tatu Ylönen 发明，他在 1995

年设计了第一版的 SSH(SSh-1)协议,用于抵御当时校园里流行的密码嗅探攻击。随后他成立了 SSH Communications Security 公司,并发布商业版的 SSH 软件。1999 年,由于开发者们希望能有一个开源的 SSH 协议软件,OpenBSD 的开发者在 Björn Grönvall's 的 OSSH 基础上开发了 OpenSSH 软件,此后 OpenSSH 成为各类 Unix 系统上标准的 SSH 协议软件。而微软视窗系统下最流行的开源 SSH 协议软件是 putty。

参考文献

1. 龚俭, 吴桦, 杨望. 计算机网络安全导论. 南京: 东南大学出版社, 2009

2. Ylonen T, et al. RFC4251: The Secure Shell (SSH) Protocol Architecture. 2006 （杨望）

SSL/TLS

SSL/TLS（secure sockets layer/transport layer security） 为网络通信提供安全及数据完整性的一种**安全协议**。它们工作在网络的传输层与应用层之间,优点是与应用层协议独立无关,应用层协议（例如 http、FTP、Telnet 等）能透明地建立于SSL/TLS 协议之上。SSL 协议是网景公司(Netscape)在推出 Web 浏览器 Netscape 的同时提出的协议,而 TLS 是 IETF 对 SSL 协议进行标准化后形成的协议(RFC2246),两者的差别极小,大多数情况下可以视为同一种协议。

SSL/TLS 协议分为两部分,即握手(handshake)规程和记录(record)规程。其中握手规程被通信双方用来进行安全的会话密钥协商;记录规程则定义了传输的格式。SSL 的通信过程也相应的分为两个阶段,握手(handshake)阶段和应用通信阶段。在握手阶段客户端和服务器双方通过握手规程进行相互的身份认证,并协商确定加密使用的算法、密钥;在应用通信阶段通信双方通过记录规程对应用协议数据使用握手阶段确定的算法和密钥进行加密通信。

在握手阶段,通信双方将通过以下流程进行密钥协商(见图 1):

（1）握手初始化阶段 ①客户端向服务器发送一个"ClientHello"消息,说明它支持的密码算法列表、压缩方法及最高协议版本,也发送客户端生成的随机数。②服务器向客户端返回一个"ServerHello"消息,包含服务器依据"ClientHello"消息选择的连接参数,以及服务器生成的随机数。

（2）身份认证阶段 客户端与服务器(根据被选择的公钥系统)交换证书和公钥,这些证书通常

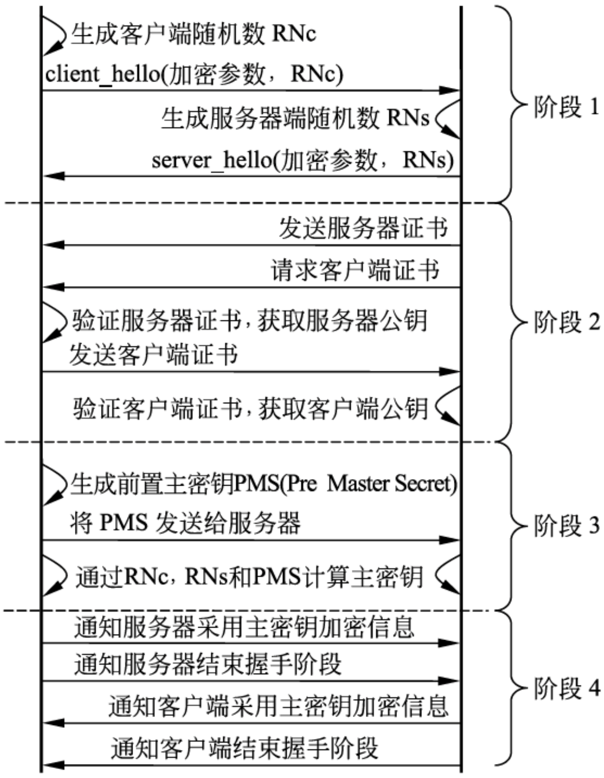


图 1 SSL/TLS 协议握手过程

基于 X. 509,但也有一些实现支持以 OpenPGP 为基础的证书。根据证书和公钥,客户端与服务器分别确认对方的身份。

（3）主密钥生成阶段 ①客户端生成随机数前置主密钥 PMS(Pre Master Secret)。②客户端向服务器发送 PMS。③客户端与服务器通过之间交互得到的客户端随机数和服务器随机数,以及客户端生成的 PMS 分别计算得出共同的主密钥。

（4）握手结束阶段 客户端和服务器分别向对方发送消息,通知对方开始使用主密钥作为之后的通信过程加密使用的密钥,同时使用主密钥加密握手结束消息,通知对方结束握手过程。

在握手阶段之后,SSL/TLS 通过记录规程封装 HTTP 等应用协议数据,记录规程会对应用协议数据进行分段、压缩和加密。在数据到达对端后,由应用协议对记录规程中封装的应用协议数据进行恢复,因此对于记录规程而言,高层的应用协议数据是透明的。

参考文献

1. 龚俭,吴桦,杨望. 计算机网络安全导论. 南京: 东南大学出版社, 2009

2. Dierks T, et al. RFC2246: The TLS Protocol. 1999. （杨望）

SVG kesuofang shiliang tuxing biaozhun

SVG 可缩放矢量图形标准(scalable vector graphics standard) 由 W3C 组织所制定的开放标准,是一种用于描述二维矢量图形的图形格式。SVG 1.0 版完成于 2001 年。2003 年,W3C 组织进一步推出两个 SVG 移动子版本: SVG Tiny 和 SVG Basic。

SVG 基于可扩展置标语言(XML),严格遵从 XML 语法,并用文本格式的描述性语言来描述图像内容,因此是一种与图像分辨率无关的矢量图形格式。SVG 具有诸多优点。首先是文件可读性强,易于修改和编辑,与现有技术可以互动融合。例如, SVG 技术本身的动态部分(包括时序控制和动画)就是基于 SMIL 标准。其次,SVG 文件还可嵌入 JavaScript(严格地说,应该是 ECMAScript)脚本来控制 SVG 对象。此外,SVG 图形格式还可方便地建立文字索引,从而实现基于内容的图像搜索。SVG 图形格式支持多种滤镜和特殊效果,在不改变图像内容的前提下可以实现位图格式中类似文字阴影的效果。SVG 图形格式可以用来动态生成图形。例如,可用 SVG 动态生成具有交互功能的地图,嵌入网页中,并显示给终端用户。SVG 面临的主要问题是如何与已经占有重要市场份额的矢量图形格式 Adobe Flash 竞争,以及 SVG 的本地运行环境下的厂家支持程度。

SVG Basic 又称 SVGB,是“Scalable Vector Graphics, Basic Profile”的简写,可以翻译为“可缩放的矢量图形标准的基本版”。它是 SVG 的一个子集,主要为掌上电脑等高端移动设备提供矢量图形显示格式。

SVG Tiny 又称 SVGT,是“Scalable Vector Graphics, Tiny Profile”的简写,可以翻译为“可缩放的矢量图形标准的微型简化版本”。它也是 SVG 的一个子集,其主要目标是为手机等低端移动设备提供矢量图形显示格式。

SVG 主要支持以下几种显示对象: ①矢量显示对象(基本矢量显示对象包括矩形、圆、椭圆、多边形、直线、任意曲线等);②嵌入式外部图像,包括 PNG、JPEG、SVG 等;③文字对象。

SVG 可以实现动态和交互功能。在 DOM 模型的基础上,SVG 开发设计人员可以利用 ECMAScript 或者 SMIL 来进行时序控制或对象的操纵。SVG 虽然是文本格式,但是 SVG 支持利用 gzip 压缩算法减少文件尺寸,压缩后的文件通常用称为“SVGZ 文件”。

由于 SVG 在实用中的诸多优点,它获得主流浏览器的支持。

参考文献

1. World Wide Web Consortium. M Media Type Registration for image/svg + xml. 22 December, 2008

2. Internet Engineering Task Force. XML Media Types, RFC 3023. January, 2001

3. http://zh.wikipedia.org/zh/SVG

(张宏鑫)

TCP/IP xieyiji

TCP/IP 协议集(TCP/IP protocol suite) 美国国防部于 20 世纪 70 年代末为 ARPANET 设计的一组通信协议,因为 IP 协议和 TCP 协议是这组通信协议中最主要的两个部分,因此整个协议集被称为 TCP/IP 协议集。这个协议集从 1983 年开始在 ARPANET 上使用,并于当年被实现在 UNIX BSD 操作系统的内核,成为 UNIX BSD 操作系统的一部分。微软公司从 Windows 95 开始也在其操作系统中提供了 TCP/IP 协议集的实现,使得 TCP/IP 协议集在联网主机中得以流行,逐渐成为 Internet 的标准协议集。

TCP/IP 协议集是一个开放的协议集合,从它的最初定义开始,其中包含的协议随着网络传输技术和应用技术的发展在不断地变化,旧的协议被淘汰,更多的新协议在增加。目前 TCP/IP 协议集中的主要协议及层次关系如表 1 所示。

表 1　TCP/IP 协议集的主要协议及其层次关系

层次名称	层内包括的协议
应用层	面向网络的:边界网关协议 BGP,路由信息协议 RIP,媒体网关控制协议 MGCP,实时传输协议 RTP,会话发起协议 SIP,简单网络管理协议 SNMP,传输层安全协议 TLS/SSL,扩展消息与存在协议 XMPP,等等。 面向用户的:域名服务协议 DNS,文件传输协议 FTP,超文本传输协议 HTTP,交换邮件访问协议 IMAP,互联网中继聊天协议 IRC,轻量目录访问协议 LDAP,网络时间协议 NTP,邮局协议 POP,简单邮件传输协议 SMTP,套接字协议 SOCKS,安全外壳协议 SSH,远程登录协议 Telnet,等等

续表

层次名称	层内包括的协议
传输层	传输控制协议 TCP,用户数据报协议 UDP,数据报拥塞控制协议 DCCP,流控制传输协议 SCTP,资源预留协议 RSVP,等等
网络层	网际协议 IP(包括 IPv4、IPv6 和 IP 协议的安全扩展 IPsec),Internet 控制报文协议 ICMP,Internet 组管理协议 IGMP/MLD,等等
链路层	地址解析协议 ARP/RARP/NDP,动态主机设置协议 DHCP,点对点协议 PPP,面向各种有线/无线通信媒体的访问控制协议,等等

参考文献

1. Comer D E. Internetworking with TCP/IP, Volume Ⅰ. 4th ed. Englewood Cliffs, NJ: Prentice Hall, 2000

2. 张尧学,等. 计算机网络与 Internet 教程. 北京: 清华大学出版社, 1999

(张尧学)

TPC jizhun ceshi

TPC 基准测试(TPC benchmarking) 按照统一的测试规范(standard specification)构造测试平台和测试程序,通过在被测试数据库管理系统上测试系统运行基准给定的工作负载,得到符合测试规范的测试报告。事务处理性能委员会(Transaction Processing Performance Council,简称 TPC)提供了联机事务处理(OLTP)、联机分析处理(OLAP)、数据中心能耗等面向数据库管理系统整体性能测评的系列测试规范。基于 TPC 基准测试指依照数据库性能测评目标,由第三方选择合适的 TPC 性能基准测试规范,完成被测系统搭建和测试过程实施,且测试过程和测试结果经过 TPC 认定的 TPC 审计者(或授权机构)验证后,其测试报告在 TPC 网站上公布的测试过程和方法。

基准性能测试一般具备下列特性:①有一个公开的包括测试目的、测试模型描述、测试环境配置要求、度量指标定义、测量计算方法、测试结果发布方式等内容的测试规范;②有符合测试规范的可执行程序或源程序,测试者使用测试数据生成程序产生测试数据库、使用测试执行程序为被测系统提供测试负载;③提供在不同的被测试系统间具有可比性的性能度量指标的测量方法或计算方法的详细说明;④在相同的测试环境下测试结果可重现,以便提供一个可比较的结果;⑤公开测试报告,公开程度一般应达到按公开的测试报告可再现测试结果。

TPC 是一个定义数据库性能测评基准的非盈利组织,成立于 1992 年,总部设在美国,其宗旨是为了制定**数据库性能测评**基准,包括性能和价格度量方法,并管理测试结果的发布。TPC 最初有 8 个会员,目前已发展到 20 家正式会员和 4 家准会员,其官方网站为 www. tpc. org。到目前为止,TPC 总共发布了 12 个性能测试基准,在用 TPC 性能基准有 5 个。TPC-C 性能基准模拟了一个大型批发销售公司货物管理业务过程,用于测试数据库管理系统在线事务处理能力。TPC-H 性能基准模拟一种拥有大量数据并且需要与在线生产数据库实现同步的**决策支持系统**,用于测试涉及大量数据和执行复杂查询的决策支持应用领域的数据库管理系统的快速响应的能力。TPC-DS 性能基准是 TPC 组织发布的第三代决策支持基准,它沿用了 TPC-H 的零售产品厂商业务场景,设计了更为复杂的查询以模拟更加丰富的决策支持功能。TPC-DS 采用了多维雪花模型进行数据建模,业务负载包括商业报表、即席决策支持、交互式 OLAP 和数据抽取式四类共 99 个查询。TPC-E 性能基准借鉴美国纽约证券交易所业务模型,模拟了一个中介公司和那些贸易、会计查询和市场研究方面的客户之间的交易。相对于 TPC-C 性能基准,TPC-E 性能基准完成了从客户/服务器(C/S)架构到浏览器/服务器(B/S)架构应用的过渡,其业务模型包含了商家对顾客(B2C)和商家对商家(B2B)两种典型的联机在线事务。TPC-Energy 性能基准是和 TPC-C、TPC-E、TPC-H 与 TPC-DS 性能基准同步,测试某个具体基准测试所消耗的能量指标。

任何用户和测试机构都可以通过网络获取 TPC 性能基准测试规范。这些基准规范是开放的,它们严格定义了被测数据库的数据逻辑、相关的事务逻辑及测试程序实现细节等,并提供了三种衡量数据库管理系统性能和价格的指标:吞吐率/响应时间、系统的总价格和价格性能比。TPC 组织不提供完整的 TPC 性能基准程序代码,任何购买者、销售者或测试实验室都可以根据规范,最优地构造出自己的测试平台。为保证测试结果的客观性,被测试者必须提交给 TPC 一套完整的测试报告,该报告必须由 TPC 授权的审核员核实,验证通过的测试结果在

TPC 网站上进行公布。事务处理性能委员会负责对测试争议进行裁决。

TPC 基准测试规范一直在更新，使其能适应计算机技术和数据库技术的发展。目前 TPC 组织正在制订 TPC-ETL、TPC-V、TPC-VMS 等数据管理系统性能基准测试规范。

参考文献

Nambiar R, Poess M, eds. Transaction Processing Performance Council: State of the Council 2012. In: Fourth TPC Technology Conference on Performance Evaluation & Benchmarking (TPCTC 2012), August 27-21, 2012

（叶晓俊　王建民）

UNIX caozuo xitong

UNIX 操作系统（UNIX operating system） 一种多用户交互式通用分时操作系统。由于它结构简练，功能强大，而且具有移植性、兼容性好以及伸缩性、互操作性强等特色，成为使用广泛，影响较大的主流**操作系统**之一，被认为是开放系统的代表。

发展简史

雏形阶段　UNIX 操作系统是 20 世纪 60 年代末由美国电报电话公司（AT&T）贝尔实验室的 Kenneth Thompson 和 Dennis Ritchie 于 1969 年实现的一种分时操作系统，最早的工作集中在文件管理和**进程**控制上，1970 年用交叉汇编方法，将该系统移植到 PDP-11 上，并提供给公司内部的专利部门用作文字处理。由于它吸取了以前的一个称作 Multics 系统的技术精华，又比 Multics 简单实用，开发者将它命名为 UNIX，这就是 UNIX 内核的雏形。

成型阶段　UNIX 设计者们继续进行功能扩展和版本更新，1972 年实现了极为重要的管道机制。1973 年 Ritchie 开发出 C 语言，它的出现是 UNIX 系统发展过程中的重要里程碑。用 C 语言改写后的第 3 版 UNIX 具有高度易读性、易移植性，为迅速推广和普及走出了决定性的一步。1974 年，“The UNIX Time-Sharing System”一文在美国杂志 CACM 上发表，引起广泛注意。最早外界可获得的 UNIX 是 1975 年的 UNIX 第 6 版。1978 年的 UNIX 第 7 版，可以看作当今 UNIX 的先驱，该版为今天 UNIX 的繁荣奠定了基础，UNIX 也步入了成型阶段。70 年代中后期 UNIX 源代码的免费扩散引起了大学和公司的兴趣，大众的参与为 UNIX 的改进、完善、传播和普及起到了重要的作用。最为著名的是美国加州大学 Berkelely 分校的 BSD 版本，其中加入了页式虚存管理、长文件名、快速文件系统、套接字、网络协议 TCP/IP 和 C-Shell 等大量先进技术，对 UNIX 的发展做出了很大贡献。

商业化阶段　1977 年 AT&T 公司开始为计算机软硬件厂商提供 UNIX 操作系统的初始设备制造商（OEM）许可证，商家开始了商业运营，许多商品化 UNIX 版本出现。比较著名的有：SUN 公司的 SUN OS 和 Solaris，Microsoft 公司的 XENIX，Interactive 公司的 UNIX 386/ix，DEC 公司的 ULTRIX，IBM 公司的 AIX，HP 公司的 HP/UX 和 SCO 公司的 UNIX 等。AT&T 公司本身则先后发展了 UNIX SYSTEM Ⅲ、UNIX SYSTEM Ⅴ，UNIX SVR4.0、4.1ES，UNIX SVR4.2 等商品化版本。到 20 世纪 90 年代，不同的 UNIX 版本已超过 100 种。

标准化阶段　商业集团的参与促进了 UNIX 技术的迅速发展与普及，但也导致了版本繁多，互不兼容的局面。因此，从 20 世纪 80 年代开始，出现了对 UNIX 标准化的工作。UNIX 用户组织最早进行此项工作，后被美国 IEEE 接受和继承，并成立了标准化工作小组，着手制定基于 UNIX 的 POSIX（可移植操作系统界面）标准，到 90 年代初已有 20 多个 POSIX 标准正式颁布与制定。与此同时，UNIX 版权拥有者 AT&T 公司也在进行努力，1984 年颁布 UNIX SYSTEM Ⅴ的界面标准（SVID）。1988 年，AT&T 与 SUN 公司宣布联合开发 UNIX SYSTEM Ⅴ第 4 版计划，拟在兼容各主要 UNIX 版本基础上，使 UNIX SVR4 成为事实上的工业标准。此举得到了 Unisys、NCR、富士通等计算机厂商的支持，但却遭到 IBM、HP 和 DEC 等厂商的不满。他们联合成立了开放系统基金会（OSF）来抵制 SVR4 计划，而 AT&T 和 SUN 等公司成立了 UNIX 国际（UI）来推动 UNIX SVR4。UNIX 分裂为互为对抗的两大集团，这种分裂与竞争促进了 UNIX 技术的进步，但也延缓了 UNIX 市场的发展。

进入 20 世纪 90 年代后，由于多处理机和分布式网络处理技术的发展，UNIX 技术也在进一步的发展。UNIX 开始支持对称多处理机、图形用户界面、分布式处理，安全性也得到进一步加强，现已演变为一种具有分布式处理能力的现代操作系统。

基本内容

UNIX 操作系统体系结构包含四个基本成分：内核、外壳（Shell）、文件系统和公用程序。

内核　是 UNIX 的基本核心。它负责调度和管理计算机系统的基本资源，包括**进程**、存储和各种设

备的管理,以及实现进程间的同步和通信。进程管理包括进程的创建、调度、执行和撤销。存储管理包括内存、外存和虚存的管理。设备管理包括**打印机**、终端机、光盘机、磁带机和磁盘机等基本外设的管理。传统 UNIX 系统把基本文件系统包括在内核中;采用**微内核**技术的 UNIX 中,内核仅提供最基本的操作系统服务,其他如网络服务、文件服务等都移到用户空间中,以提高内核效率和便于内核移植。

文件系统 负责组织并管理数据资源。UNIX 文件系统采用树形层次结构,是一棵有根倒向树。最上端是根目录,第二层通常包括 etc、bin、lib 和 user 子目录。目录的层次可以不断扩充,而树枝是子目录,树叶为文件。可以通过路径名来访问目录和文件。早期 UNIX 仅支持一种字符串格式文件,目前 UNIX 已能支持包括"虚拟文件系统"在内的 10 多种文件系统,可以适应各种不同应用的需要。

外壳(Shell) 是一种**命令式语言**及其解释程序,命令语言是 UNIX 早期的用户界面。由于 UNIX 有管道(pipe)机制,而且提供了功能齐全的命令,加上 Shell 自身具有控制功能的语句成分,使得 Shell 命令语言功能相当强大,用户可在更高层次上进行程序设计,提高程序开发效率。Shell 的常用版本有 Bourne-Shell、C-Shell、Korn-Shell。WK-Shell 是具有图形开发能力的最新版本。

公用程序 或称工具软件,是 UNIX 系统提供给用户使用的常用标准软件,其内容相当丰富,包括编辑工具、管理工具、网络工具、开发工具、保密与安全工具等。目前所说的 UNIX 已经不再单纯指 UNIX 操作系统,因为,它提供了丰富的工具软件,已经发展和演变成为一个功能强大的软件开发和运行环境。

主要特色

结构简练 以精巧的文件系统为代表。采用树形目录结构,文件的查找、增、删、改十分方便。文件系统可装卸,便于用户裁剪。外围设备都定义为特殊文件,统一以文件方式进行处理,简洁明了。

功能强大 除了常用功能外,UNIX 还首创了 pipe,它能将许多小功能片断连接组装、巧妙结合成复杂功能的软件工具。UNIX 也是最早具有创建异步进程能力的系统,早在 20 世纪 70 年代就实现了多用户多任务功能。此外,UNIX 提供一系列的网络通信工具和协议,适用性好、使用广泛。

易移植性好 这一特色源于 **C 语言**和源代码开放政策。UNIX 系统 90% 以上代码用 C 语言编写,因此,易移植性好。到 1984 年,它已被移植到 70 多种计算机系列上。

可伸缩性和互操作性强 前者指在范围很宽的性能和配置的硬件上运行的能力;后者指在不同厂家的机器上运行和通信的能力。这两点是开放系统的基本特征,也是 UNIX 的重要特色,它从笔记本电脑到巨型机上都可以运行。迄今,已安装和运行在 100 多家计算机厂商的硬件平台上,这是其他操作系统无法比拟的优势。

完善的安全机制 由于 UNIX 基于多用户、多任务环境,设计了周到的安全机制,包括对用户的管理、对文件使用权限的管理、网络通信安全管理等。当今在世界各国许多关键性行业的信息化管理系统大都采用 UNIX 系统支撑。

发展趋势

UNIX 系统目前安装数量超过 500 万套,用户数达到 3000 万,已成为一种主流操作系统。从总体上看,UNIX 操作系统的主要发展趋势是统一化、标准化和不断创新。1993 年"公共开发软件环境(COSE)"组织成立,标志着主要 UNIX 厂商的联合和 UNIX 系统统一化的开始。同年,Novell 公司从 AT&T 公司购得的 UNIX 商标权无偿交给开放系统标准化组织 X/OPEN,这表明 UNIX 商标不再受某一厂商控制。在该组织的推动下,UNIX 的两个重要标准 Spec. 1170(标准应用程序界面)和 CDE(公共桌面环境)已于 1995 年正式颁布,为 UNIX 的统一化、标准化打下了重要基础。总之,由于 UNIX 的开放性,使它的发展充满活力和生机,与 UNIX 有关的新技术和新产品会不断涌现,UNIX 正是在这种既竞争,又协作的环境中不断发展和前进的。

参考文献

Muster J. UNIX 和 Linux 权威教程. 王玉馨,郑建超,等译. 北京:清华大学出版社,2003

(贾耀良　费翔林)

VAL yuyan

VAL 语言(VAL language) 典型的**数据流语言**。VAL 是 Value - oriented Algorithmic Language 的缩略语。1979 年由美国麻省理工学院(MIT)的 J. B. Dennis和 W. B. Ackerman 提出。它是一个小型研究性语言,其设计目标是为调整计算提供一个实验环境,以便人们了解数据流的计算过程。VAL 只提供了能表达算法固有并行性的最基本部分。它遵

循以下两条设计原则:①提供隐式并行,任何指令级的并行均由计算机系统本身自动开发,而不需依赖语言的任何并行机制。VAL 的完全函数性支持了隐并行目标;②能有效转换为 DFG 和产生高效的程序。该语言简单明了,有助于程序人员了解计算系统的内部活动,进行错误处理、调试和效率分析。

VAL 的主要成分可分为说明部分和活动部分。详见表 1。

表 1　VAL 成分

说明部分	基本类型	·整型·布尔型
		·实型·字符型
	构造类型	·数组·记录
		·择一
活动部分	表达式	
	函数	

VAL 具有很多强类型语言的特征,其值的类型和作用域都必须在程序中严格说明。每一类型,除了其定义值外,还包括相应的错误值。一旦出错,能迅速中止程序执行并作出处理。数组类型定义仅说明元素类型,不说明数组界限,界限直至构造数组值时才确定;择一类型的值在执行过程的不同时刻可有不同的类型。VAL 程序由一组模块组成,每一模块包含一个外部函数,可供其他模块调用。VAL 只有函数,没有过程。函数可嵌套。

为了适应数据流的需要,VAL 舍弃了若干强制式语言的特性(如变量、GOTO 语句),还提供了有助于描述数据流算法和各种表达式。

VAL 与 FP(函数式程序设计)在纯函数性、表达式、无副作用和程序代数规则等方面很相似,但二者也有明显的区别:VAL 有类型,FP 没有;FP 有函数型运算,VAL 没有。VAL 的主要缺点是:未提供通用的输入输出手段,缺乏递归性,未说明用什么方法实现结果收集。

参考文献

1. Ackerman W B, Dennis J B. VAL — a value-oriented algorithmic language: preliminary reference manual. Tech. Rep. TR — 218, Computation Structure Group, Laboratory for Computer Science. Cambridge, MA: MIT Press, 1979

2. McGraw J R. The VAL language: description and analysis. ACM TOPLAS, 1982, 4(1)　(郭浩志)

VLIW chuliji(qi)

VLIW 处理机(器)(very long instruction word processor)　一种超长指令字处理机(器)。在 VLIW 处理机的一条指令(通常称为超长指令)中包含有多个操作字段,每个操作字段的功能相当于一般处理机中的一条指令,一条超长指令能够独立控制多个功能部件同时工作。

VLIW 处理机设计思想由美国的 J. A. Fisher 于 1981 年首先提出。J. A. Fisher 等人领导的 Mutiflow 公司研制了世界上第一台 VLIW 处理机 TRACE28/300,该处理机的一条超长指令中最多能够有 28 条可以同时执行的指令。

提出 VLIW 处理机的主要目的是要充分开发程序中的指令级并行性,并简化硬件结构。目前,采用指令级并行方式的处理机主要有超标量处理机、超流水线处理机和 VLIW 处理机等三种。**超标量处理机**依靠设置多条指令流水线,并通过同时发射多条指令来提高处理机的运算速度;而**超流水线处理机**则通过分时使用同一条指令流水线的不同部分,采用并发并行方式来提高处理机的运算速度。与超标量处理机和超流水线处理机相比,VLIW 处理机的主要特点如下:

(1)采用**显式并行指令计算**(explicitly parallel instruction computing, EPIC)方式　VLIW 处理机主要依靠并行编译器显式开发程序中的指令级并行性,硬件对编译器开发指令级并行提供强有力的支持。在 VLIW 处理机上运行的目标程序可以看成是一个二维指令矩阵,指令矩阵中每一行上的所有指令组成一条超长指令,它们之间没有数据相关、控制相关和功能部件冲突,可以在 VLIW 处理机上同时执行。超标量处理机和超流水线处理机通常采用隐式并行方式,在这两种处理机上运行的程序是一维线性的指令序列,每条指令中一般只包含一个操作。在程序运行过程中,超标量处理机和流水线处理机主要依靠硬件在一个不大的指令窗口中寻找指令级并行性,找出几条与流水线中所有正在执行的指令之间没有数据相关、控制相关及功能部件冲突的指令来并行执行。

(2)指令级并行度高　超标量处理机和超流水线处理机的指令级并行度一般为 2 左右,通常不超过 4,而目前多数 VLIW 处理机的指令级并行度在 4 至 8 之间,甚至超过 10。VLIW 处理机的指令级并行度高于超标量和超流水线处理机的主要原因是:由于在 VLIW 处理机中通过并行编译器来开发程序

中的指令级并行性,可以在一个循环、一个函数、甚至整个程序中寻找指令级并行性,并且可以采用软件流水、循环展开等高并行度方法充分开发程序中的多种并行性。

(3) 硬件结构规整、简单　VLIW 处理机主要由很规则的寄存器、存储器、运算部件和数据通路等组成,不规则的控制器比较简单。另外,由于 VLIW 处理机主要依靠编译器开发程序中的指令级并行性,不需要复杂的指令并行调度窗口及多发射机制等。

(4) 编译器的实现难度大　VLIW 处理机的并行编译器主要依靠指令级并行算法、数据相关性分析算法、寄存器分配算法等来显式开发程序中的指令级并行性,从而提高处理机的运行速度。要研制指令级并行度高的编译器难度很大。

许多计算机公司、著名大专院校都研制了自己的 VLIW 处理机,市场上也出现了多种 VLIW 处理机,其中,大部分是嵌入式、DSP、JAVA 虚拟机等专用处理机,其主要原因是,针对专用应用领域的 VLIW 编译器相对容易实现。已经推向市场的通用处理机有:Intel 和 HP 公司联合推出的安腾(Itanium)处理机,IBM 公司推出的 DAISY 处理机,Transmeta 公司推出的 Crusoe 处理机等,Crusoe 处理机已经大量应用于**笔记本计算机**中,其功耗很低。其中,安腾处理机可以同时发射 6 条指令,它有自己独立的系统软件和应用软件。

为了使 VLIW 处理机能够与其他通用处理机实现二进制兼容,IBM 公司推出了开放源代码 DAISY,它不仅可以实现 IBM 的 VLIW 处理器与 x86 处理机之间的二进制兼容,还可以实现 PowerPC、S/390、IBM 的 Java 虚拟机与 VLIW 处理器之间的二进制兼容。Transmeta 公司推出了代码映射软件,它可以使 Transmeta 公司的 VLIW 处理机 Crusoe 与 x86 处理机之间实现二进制兼容。　(汤志忠)

VLSI suanfa

VLSI 算法(VLSI algorithm)　一类适用于大规模集成电路实现的算法。VLSI 算法的两个重要指标是:时间复杂度(并行计算过程所耗费的时间)和面积复杂度(并行计算过程在超大规模集成电路上实现时相应计算结构所占的面积,这里的计算结构是指由大量处理器按照一定方式连接起来的结构,它可以抽象成一个图)。问题求解的并行计算过程与支持这一过程的计算结构相结合是此类算法的特点。VLSI 算法研究,主要是通过对问题计算规律的分析,挖掘出问题计算过程中可能存在的并行性,进而找出能充分利用这种并行性的并行算法以及支持这一并行计算过程的计算结构,最后分析算法的时间复杂度和面积复杂度。

用于 VLSI 算法的计算结构有多种,例如线性阵列和环、二维网格和环绕网、心动阵列、星图、多维网格、树、蝶形结构、洗牌交换网和超立方体及其若干变型结构等。图 1 示出了几种主要的计算结构。

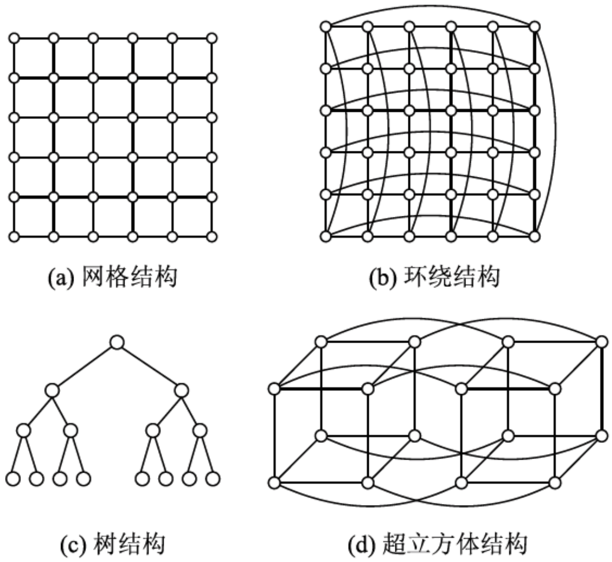


图 1　几种主要计算结构

对 VLSI 计算结构的研究工作主要有如下几个方面:

一是关于实用计算结构的研究。例如,为解决二维网格上结点间通信线路长(对 $N \times N$ 网格来说是 $O(N)$)和行列统计运算时(一般要 $O(N)$时间)等问题,引进了树网结构;为解决树结构中结点通信不畅,特别是根结点附近信息堵塞严重问题,提出了 *X*-树结构(树中同一级结点间加进水平通信线)和胖树结构(越接近根结点,通信线路和结点中处理通信的部件越多);为解决超立方体结构中每个结点通信线较多的问题,提出了 CCC 网结构(在这种结构中,每个结点只有三个通信线);等等。

二是关于各种计算结构性质的研究,主要包括哈密尔顿性质、可嵌入性及可诊断性等。

计算结构中的哈密尔顿路径(若存在)可用于双路径、多路径的多点路由算法,以减少或避免拥塞和死锁。

图嵌入在计算结构中有多种应用,如体系结构模拟、处理器分配等。一个图嵌入是由一个客图的

顶点到一个主图(计算结构)的顶点的映射。常用的客图有线性结构(包括环)、二维网格结构(包括环绕网)、多维网格、树结构等。

一个大规模系统中,由于存在大量处理器和链路,处理器或链路发生故障是不可避免的。因此,故障诊断是保证系统结构可靠性的一种重要手段。常用的诊断模型有 PMC 模型和比较模型。

常见的 VLSI 并行算法种类有:①基本数值计算算法,如数字表达式的并行计算、矩阵相乘、方阵的分解与求逆等;②数字信号处理中数值计算算法,如卷积、数字滤波、傅里叶变换等;③非数值计算算法,如排序算法、字符串模式匹配等;④图论算法,如图的传递闭包算法、图的连通分量算法、图的最短路径算法、图的着色算法等。

在 VLSI 算法中,还有一类算法曾受到人们的特殊关注,这就是心动(systolic)算法。心动算法又称脉动算法、搏动算法。心动算法的相应计算结构称为心动阵列。

对于 VLSI 算法的评估,主要在于算法的执行时间和芯片面积两个方面。执行时间是指算法从开始到结束所经过的时间,一般包括两部分:①数据从一个 PE(processor element)经由计算结构到另一个 PE 所需的时间;②数据在一个 PE 内的算术、逻辑等运算时间。由于 VLSI 并行计算的同步性与规则性,通常将一次数据传送与一次计算的时间合起来称为一个单位时间(time unit)或计算步(computational step),并以此作为度量算法执行时间的单位。在 PE 互连简单且规则的情况下,芯片面积的大小也可以近似地用 PE 个数来衡量。

设某 VLSI 算法的执行时间为 T,芯片面积为 A,VLSI 算法评估的标准是 A 与 T 的函数 $f(A,T)$。不同的 f,可以得到不同的评估参数。设 n 表示计算问题的规模,$P(n)$ 表示处理器数目,以它近似衡量芯片面积 A。

一些基本操作(例如加、减、乘、除和比较运算等)的时空复杂度是各种 VLSI 算法复杂度分析的基础,这方面已有一批成果。通常 k 位数加法的时间空间复杂度均为 $O(k)$,但也有时间复杂度为 $O(\log k)$、空间复杂度为 $O(k\log k)$ 的加法器。各种基本结构的 VLSI 布局,直接影响 VLSI 算法的面积复杂度。采用 H-树布局方法,可使 N 个叶子的完全二叉树的面积由通常的 $O(N)\times O(\log N)$ 降到 $O(N)$。关于下界的研究有助于对问题并行计算复杂性本质的了解,也有利于对问题的 VLSI 算法性能作出评估。由于 VLSI 算法的特殊性,必须同时考虑时间 T 和面积 A 两个因素。经常研究的子因素有 A 下界、AT 下界和 AT^2 下界。已经证明,n 个 k 位数的排序,对于 $k\leqslant \log n$、$\log n<k<2\log n$ 和 $k\geqslant 2\log n$ 情况下的 AT^2 下界分别是 $n2^k$、$n^2\log^2(2^{k+1}/n)$ 和 $kn^2\log n$。

参考文献

1. Ullman J D. Computational aspects of VLSI. Computer Science Press, 1984

2. Kung S Y. VLSI array processors. Englewood Cliffs, NJ: Prentice Hall, 1988

3. 陈国良, 陈崚. VLSI 计算理论与并行算法. 合肥: 中国科学技术大学出版社, 1991　(樊建席)

Web fuwu

Web 服务(Web services, WS)　一种面向服务的架构技术。它通过标准 Web 协议提供服务,使不同平台的应用服务或电子设备通过网络进行通信与互操作。根据国际标准化组织 World Wide Web Consortium(W3C)的定义,Web 服务是一个软件系统,支持通过**计算机网络**进行机器间交互操作。Web 服务提供接口,采用机器可处理方式(如 Web 服务描述语言(Web Services Description Language, WSDL))对服务进行描述,采用通信协议(如 Web 服务消息协议(Simple Object Access Protocol, SOAP))描述服务之间的交互或消息传递方式。Web 服务具有如下特征:①分布式松耦合　Web 上的资源和服务提供者分布在网络上不同管理域,并且是松散耦合的。②互操作性　Web 服务的界面、消息通信、事务处理、工作流和安全机制都采用标准规范的协议和约定,支持应用系统间的互操作。③平台无关性　Web 服务的规范协议都基于可扩展置标语言(extensible markup language, XML),能够屏蔽不同软件平台差异,不依赖具体编程语言和系统平台。

Web 服务技术始于 20 世纪 90 年代初,是伴随着分布式计算技术的发展而成长的。其发展过程可概括为三个阶段:①孕育阶段　以 XML 技术为基础,任何文档和数据可以转换成统一的 XML 格式进行描述和交换,实现跨越互联网协议传输。②成熟阶段　2000 年以后,人们为了实现基于**互联网**和开放、自由的通信框架以及实现异构系统间交互,提出了 Web 服务概念。国际标准化组织 W3C 提出 SOAP、WSDL 和 UUDI(Universal Discovery Description and Integration) 等 3 个 Web 服务标准和规范,标志

着 Web 服务技术开始成熟。③发展应用阶段　近年来,随着 Web 服务标准和技术的成熟,Web 服务被互联网应用、**服务计算**、**云计算**以及各行各业广泛采用,并形成了 Web 2.0 和 Web 3.0 等。

Web 服务的基本核心定义包括 WSDL、SOAP 和 UDDI 等协议。WSDL 是 XML 格式的文档,用以描述服务接口访问方式和使用协议的细节。SOAP 是基于 XML 可扩展的消息信封格式,通常绑定一个传输协议,如互联网上广泛采用的 HTTP 或 HTTPS 协议,但有时也用 SMTP 或 XMPP 等。UDDI 是一个用来发布和搜索 Web 服务的协议,应用程序可借由此协议在设计或运行时找到目标 Web 服务。Web 服务提供者将 WSDL 描述的 Web 服务发布到 UDDI 服务注册器;Web 服务消费者从此注册器发现所需 Web 服务,通过发现的 Web 服务 WSDL 描述获知如何使用 Web 服务,从而引用所需的 Web 服务。Web 服务的调用方法有多种,最常用的有:远程过程调用(remote process call,RPC)、面向服务的架构(service-oriented architecture, SOA)、表述性状态转移(representational state transfer, REST)等。

为了扩展 Web 服务能力,一些标准规范已经或正在被国际标准组织开发。这些标准通常被冠以 WS(Web service),主要常用标准包括:WS 安全(WS-security)、WS 可靠性(WS-reliability)、WS 可靠消息(WS-reliable messaging)、WS 寻址(WS-addressing)、WS 事务(WS-transaction)、WS 资源框架(WS-resource framework)等,还有 WS-resource、WS-resource property、WS-notification 规范等。

除了 Web 服务规范和协议标准化问题外,Web 服务核心支撑技术和基于 Web 服务的解决方案关注的研究问题包括:

(1) Web 服务事务处理　在松散、分布式环境中,如何进行相互协同,利用已有的事务处理机制实现 Web 事务处理。

(2) Web 服务发现与选择　针对 Web 环境下大量的 Web 服务,如何准确、高效地发现和选择所需的 Web 服务。

(3) 服务组合　如何利用已有资源,按照一定的粒度重用 Web 服务,进行 Web 服务组合,自动业务服务流程,形成复杂服务应用系统来满足不同应用需求。

(4) 基于语义的 Web 服务　结合语义信息,使得 Web 服务更能体现用户预期目标和制约条件,得到更精确的结果,从而改善服务质量。

(5) Web 服务评价与优化　如何描述和评价服务质量、Web 服务组合代价以及服务执行结果正确性验证,权衡影响 Web 服务质量的各种因素,最优系统总体性能。

(6) Web 服务质量(QoS)管理　高效服务质量管理策略探索及服务质量代价模型建立,是改善服务质量必须解决的问题。

(7) Web 服务安全　在开放的 Web 环境下,选择制定有效的 Web 服务应用系统安全、认证和加密等策略。

Web 服务自动开发方法有两种:自底向上方法和自顶向下方法。采用自底向上方法,开发人员先利用某种编程语言实现对象类,再用 WSDL 文档生成工具将对象类中的方法表示为 Web 服务操作,可大大简化开发过程;采用自顶向下方法,开发人员先编写描述 Web 服务的 WSDL 文档,再用代码生成工具产生实现服务操作的目标类。此方法相对难些,但是可得到清晰的 Web 服务设计。

除了 Web 协议集合、应用实体集合外,Web 服务还是一个集应用逻辑、网络技术、工作流管理、知识表示、逻辑推理、安全保密、信息集成和先进计算模式等技术为一体的新兴应用模式。在这一新模式驱动下,Web 服务及其应用未来的发展趋势为:①Web服务将更加智能。Web 服务与语义和本体论的结合,使得 Web 服务发现、组合更加自动和智能。②Web 的内容更加丰富。随着 Web 2.0、Web 3.0 和 Web n.0 的发展,Web 服务将集成更多不同内容来源,使其服务内容更加丰富和及时。③Web 服务质量更加可靠。Web 服务应用云计算等先进的计算模式和支撑技术,使其服务功能更加强大,服务质量更加可靠。④Web 服务更加安全可信。成熟的 Web 安全机制,以及先进的 Web 服务安全访问控制和数据加密技术,将使 Web 服务更加安全可信。更重要的是,Web 服务技术使得应用程序开发技术从以操作系统为中心的应用程序组织模式扩展到以网络为中心的组织模式,在视野上从本地扩大到了全球;使得数据共享方式从原来的人-人、人-机器模式发展到机器-机器(软件-软件)模式,将促进在全球范围实现软件和信息自动化生产的大工业产业模式的发展。

参考文献

1. W3C Working Group. Web Services Architecture. http://www.w3.org/TR/ws-arch/

2. IBM. Secure, Reliable, Transacted Web Serv-

ices. http://www. ibm. com/developerworks/webservices/library/ws-securtrans/

（臧天仪）

Web wajue

Web 挖掘（Web mining）　对 Web 数据的挖掘（参见**数据挖掘**）。Web 数据包括 Web 页面、隐藏在页面后的数据库数据，以及网站日志。此类数据的特点是规模大，常通过超链接发生关联，具有半结构化和非结构化性质。

Web 挖掘技术可以大致分为三类：内容挖掘、结构挖掘、使用挖掘。

（1）Web 内容挖掘是指通过对来自不同网站、由不同组织或个人发布的网页或数据库内容进行分析，为搜索、查询、个性化推荐或其他应用提供支持。Web 内容挖掘技术包括页面分类、页面聚类、查询接口分类等。由于 Web 页面的本质是文档，因此，Web 内容挖掘技术大量借鉴和使用了信息检索和**文本挖掘**的有关技术。另一方面，Web 内容的数据量巨大，并且在处理时每个文档通常用一个高维向量表示，为了提升相似度比较的效率，通常对 Web 内容数据进一步使用位置敏感散列（LSH）进行预处理。

（2）Web 结构挖掘是指利用 Web 上的链接信息或者所隐含的社交网络对其结构进行分析。Web 结构挖掘可用于对页面重要性进行分析，从而提升搜索的准确性；也可用于发现 Web 社区或对页面进行分类或聚类。Web 结构挖掘可以是与查询无关的，如 PageRank 算法对每个 Web 页面计算其全局的重要程度；也可以是与查询有关的，如 HITS 算法针对每个查询计算各个页面的权威程度和索引重要程度。与查询有关的 Web 结构挖掘通常能够更准确地获得分析结果，也易于针对用户进行个性化定制，但是往往需要耗费更多的计算资源。此外，Web 结构挖掘还包括对于 Web 结构生成规律的研究和分析，研究结果可用于生成小规模但具有 Web 结构特征的数据集，它可用来测试 Web 结构挖掘算法的有效性。

（3）Web 使用挖掘是指对用户的 Web 访问和使用方式进行分析。分析的对象通常以网站日志和用户 cookie 为中心，分析的目的包括了解用户需求、优化网站或广告布局、优化搜索排名、个性化推荐等。Web 使用挖掘技术包括用户会话发现、频繁访问路径和访问模式的发现、用户搜索意向发现等。

在实施时，为了有效和及时地处理海量的 Web 数据，通常需要为 Web 挖掘设计相应的能在大规模集群计算机上运行的并行算法，以保证在充分利用分布式存储的同时实现处理的并行性，提升处理效率。

Web 挖掘技术在网站优化、**电子商务**、搜索引擎优化、垃圾信息检测等方面得到了广泛的应用。它在信息服务的个性化推荐与推送、互联网广告等**互联网**新兴应用中所起的作用也越来越重要。

参考文献

1. Soumen Chakrabarti. Mining the Web: Discovering knowledge from hypertext data. Morgan-Kaufmann Publishers, 2002

2. Liu B. Web data mining: Exploring hyperlinks, contents, and usage data. Heidelberg: Springer, 2007

3. Rajaraman A, Ullman J. Mining of massive datasets. Cambridge University Press, 2011

（钱卫宁　周傲英）

Web 2.0 yingyong

Web 2.0 应用（Web 2.0 applications）　一种促进**互联网**环境中人与人之间信息交换和协同合作的互联网应用新模式。相对于 Web 1.0 而言，Web 2.0 更加以用户为中心，将众多用户所创造的信息进行收集、混合与整理，形成新的有价值的信息，进而以各类技术手段通过互联网向用户发布和提供用户使用。另一被广泛认可的定义是：“Web 2.0 是一个构建在知识上的环境，使人与人之间交互而产生的内容，经由面向服务的架构中的程序在这个环境中被发布、管理和使用。”Web 2.0 是相对于 Web 1.0 应用的一个概念，被看作是互联网核心理念和思想体系的一次升级换代，由原来的自上而下的由少数资源控制者集中控制主导的互联网应用模式转变为自下而上的由广大用户集体智慧和力量主导的互联网应用模式。互动、分享、关系是 Web 2.0 的核心概念；由于用户参与所带来的开放性、自由、协同智能被认为是 web 2.0 的关键属性。Web 2.0 包含了我们经常使用到的服务，例如博客、播客、维基、P2P 下载、社区、分享服务等。

20 世纪 90 年代，Web 1.0 随着互联网的发展而产生，被称为“以数据为核心”的网络。Web 1.0 主要以静态 Web 网站和一系列页面构成，每个网站上提供的资源由某特定的公司、组织或个人所构建和发布，用户对资源进行检索、查看和其他简单的操

作。Netscape 是 Web 1.0 典型代表。21 世纪初,随着互联网用户的爆发式增长,这种自上而下构建信息的 Web 1.0 应用模式已经无法适应人们对信息的创造与获取需求。互联网应用的主导权从少数组织转移到了成千上万的普通大众用户手中,由互联网用户靠集体智慧来创造更多的资源和信息。这些信息通过社会关系网络在互联网上进行快速和大范围的传播与分享,进而创造出更多的信息,由此产生了 Web 2.0 应用模式。这是一种“以人为中心”的网络。Web 2.0 由 Tim O'Reilly 在 2003 年首次提出,他对 Web 2.0 应用的特征归纳出了如下观点:互联网作为平台、集体的智慧、数据的智能化处理、软件就是服务(SaaS)、传统软件周期的终结、轻量级编程模式、超越单一设备层的软件、丰富的用户体验等。

从技术上看,Web 2.0 具有分布式结构、松散耦合、平台独立性、开放应用程序设计接口(API)、支持 **Web 服务**、富界面应用、内容协同等技术特征。Web 2.0 应用体系结构集中体现了其技术内涵,分为三个层面:

(1) 信息创造与获取层(get) 广泛收集互联网上存在的各类公共数据(例如地图信息、电视节目信息、天气信息等)、由用户创造出来的各类信息(例如 blog、Wiki、图片、视频、书评等)、由某些公司/组织聚集起来的信息(例如 Google 搜索、图书排行榜等)、各类元数据(tag)等。该层次的主要技术包括:支持用户创造数据的平台和工具、多用户间协作与交互技术(例如即时通信、博客、微博、Wiki 等)、开放的 API 接口(例如 Google API、Facebook API 等)、数据语义标注(例如 Tag 技术等)、Semantics Web 构建技术等。

(2) 信息混合与整理层(Remix) 采用聚合、标注、转换、过滤、索引、映射、排序等手段,对各类信息进行混合、整理和重新组织,从而形成更有价值的信息;例如 PHP、Ruby on Rails、Perl、Python 等新的程序设计语言,支持信息共享的 XML、RSS 和 JSON 信息表示等。该层次的主要技术包括:各类数据的高效率自动获取技术、海量数据的分布式存储和高效率查询技术、海量数据的信息检索和分析技术、海量数据挖据和知识发现技术、语义自动化处理技术、数据混合整理结果的高效率组织和表示等。

(3) 信息发布层(deliver) 通过丰富的用户界面展现形式(例如 Ajax、Flash/Flex、DHTML、HTML5、Widget 等),以**可视化**、个性化方式将信息推送给更多的用户;并支持用户使用传统 PC、手机等各类设备随时随地获取和处理信息,提供丰富的客户体验。该层次的主要技术包括:信息的发布-订阅和推送机制、信息可视化展示与富客户端的 Web 页面开发技术、支持各类异构设备的 Web 页面开发技术、跨技术的标准化协议等。

近年来,Web 2.0 主导了互联网应用,创新型应用层出不穷,极大推进了互联网和 IT 产业的发展。典型的 Web 2.0 应用包括:社交网络服务(Facebook、人人网等)、博客、维基百科全书(Wikipedia)、视频共享(YouTube)、图片共享(Flicker)、微博(Twitter、新浪微博等)、地图服务(Google Maps)、地理位置服务(LBS)、内容源(RSS)、社会化书签(Del. icio. us)等。

Web 2.0 未来发展趋势可用 SoLoMo 表示,即 social(社会化)、local(本地化)、mobile(移动化)。社会化是指将现实中人与人之间的社交网络映射到了互联网虚拟空间中,基于社交网络展开成员之间的交流/共享/传播,进而支持各类 Web 2.0 应用服务;本地化是指通过无线网络(GSM、3G、WiFi)或外部定位方式(GPS、RFID)获取移动终端用户的位置信息,进而为用户提供各类与位置相关的增值服务应用;移动化是指用户可借助各类移动终端接入服务系统,随时随地访问云端的服务。归纳起来,使用户可以在任意时间和地点进行内容的创造、分享、聚合、获取,这是一种深层次的 Web 2.0 发展趋势。

参考文献

1. Tim O'Reilly. What is Web 2.0. http://oreilly. com/web2/

2. Amy Shuen. Web 2.0 策略指南. 赵俐,盛海艳,等译. 北京:机械工业出版社,2009

(王忠杰　王显志)

WiMax

WiMax(Worldwide Interoperability for Microwave Access) 全球微波互联接入。WiMAX 也叫 802·16 无线城域网或 802.16。WiMAX 是一项宽带无线接入技术,能提供面向互联网的高速连接,数据传输距离最远可达 50 km。WiMAX 具有 QoS 保障、传输速率高、业务丰富多样等优点。WiMAX 的技术起点较高,采用了代表通信技术发展方向的 OFDM/OFDMA、AAS、MIMO 等先进技术,随着技术标准的发展,WiMAX 逐步实现宽带业务的移动化,而 3G 则实现移动业务的宽带化,两种网络的

融合程度会越来越高。　（周正）

Windows caozuo xitong

Windows 操作系统（Windows operating system）　由美国 Microsoft 公司开发，支持多道程序运行，具有图形界面环境的**操作系统**。Windows 最初是作为对 **DOS 操作系统**的图形化扩充而推出的，已推出多个版本。它的多任务图形界面以及统一的应用程序接口使得在 Windows 环境下运行的应用程序的操作大为简化。Windows 获得了**微型计算机**操作系统的垄断地位。它在服务器软件市场也有相当建树。

发展简史

Microsoft 公司在 1983 年开始研发 Windows，其最初目标是在 DOS 操作系统的基础上提供一个多任务的图形化用户界面，并希望它能够成为基于 Intel x86 微处理芯片计算机上的标准操作系统。

继 1985 年和 1987 年分别推出 Windows 1.0 版和 Windows 2.0 版后，Microsoft 公司于 1990 年 5 月推出了 Windows 3.0。Windows 3.0 对 Windows 的内存管理、图形界面做了重大改进，使图形界面更加美观并支持虚拟内存，它以压倒性的商业成功确定了 Windows 系统在个人计算机领域的垄断地位。

随后，Microsoft 公司于 1995 年推出新一代操作系统 Windows 95，它对 Windows 3.x 版做了许多重大改进，包括：更加优秀的、面向对象的图形用户界面；全 32 位的高性能的抢先式多任务和多**线程**；内置对 Internet 的支持；更加高级的多媒体支持（声音、图形、影像等）；即插即用；32 位线性寻址的内存管理和良好的向下兼容性等。它可以独立运行而无须 DOS 支持，是操作系统发展史上一个里程碑式的作品。

在 Windows 95 取得巨大成功之后，Microsoft 公司继续对桌面版的 Windows 进行升级，Windows 98 在操作界面、联机帮助及辅助工具向导等方面都有了很大改进。它增加了用于自动检测硬盘、系统文件和配置信息的系统工具，内置了大量的硬件设备驱动程序，融合了当时最新的多媒体技术、网络技术。Windows Me 则是 Windows 98 的继续升级版，对用户提供更加强大的多媒体功能、高集成度的网络和更加友善的用户界面。

Windows 家族中另一重要的分支是 Microsoft Windows NT，它是由 Microsoft 发行的面向高端的操作系统。Windows NT 作为全新设计的操作系统，与原先支持个人应用的 Windows 有根本的区别。它采用客户-服务器与层次式结合的模型，体现了**微内核**结构操作系统 Mach 的思想，可以在从桌面系统到大型多处理器的网络服务器等一系列机器上运行。Windows NT 支持多进程并发，有较强的内置网络功能，系统安全性也达到较高水平。它所包含的 Win 32，Win 16，MS DOS，OS/2 和 POSIX 子系统提供了优越的应用程序兼容性，这一点是此前的任何操作系统无法相比的。

Windows 2000 是新一代个人计算机的商务操作系统。它建立在 NT 技术之上，具有高可靠性，高扩展性和业务优势，它通过简化系统管理降低了操作消耗，是一种适合从最小的移动设备到最大的电子商务服务器新硬件的操作系统。

Windows XP 是第一个把消费型操作系统和商业型操作系统融合为统一系统代码的 Windows，它结束了 Windows 两条腿走路的历史，是第一个既适合家庭用户，也适合商业用户使用的新型 Windows。

最新的 Windows 产品是 Windows 2003，它提供了联网、**消息传递**、集群、**数据库**到**电子商务**互联网（Web）站点以及文件和打印服务器等操作系统基础设施，具有高可靠性和高性能以及优异的商业价值。

到目前为止，Microsoft 推出了支持从**个人数字助理**、移动电话、接触式屏幕设备（如 Windows CE）到个人计算机、工作站、大型多处理器等的一系列 Windows 操作系统。

Windows 系统的构成

依据其提供的系统服务，Windows 系统主要由以下 3 个基本模块组成：

（1）内核　内核实现对计算机资源的管理，并提供系统服务和 Windows 的多任务管理，支持 Windows 应用程序所要求的低级服务，如动态内存分配，**进程**管理和**文件**管理等功能。

（2）图形设备接口（GDI）　图形设备接口是一组图形设备驱动程序和库，是 Windows 图形功能的核心，它支持字体、绘图原语和用户显示及打印设备的管理。在此基础上，可实现 Windows 系统与设备无关的图形界面，并提供图形编程接口。

（3）用户模块　用户模块实施对窗口的管理，且提供编程接口和外壳（Shell）功能。Windows 向用户提供两种类型的 Shell：程序管理和文件管理，它们在形式上是一个窗口，用户对 Windows 的各种操作，都是在 Shell 窗口下进行的。

随着 Windows 的不断发展，Windows 系列的新

产品在基本保持原结构的基础上，对上述 Windows 系统的基本功能模块作了相当程度的扩充和改进，以 Windows 2000 为例，从性能上的改进主要体现如下：

（1）可扩展性　Windows 2000 的可扩展性依赖于它的环境子系统。环境子系统向应用程序提供运行环境（操作系统功能调用接口），Windows 2000 有 3 个环境子系统：Win32、POSIX 和 OS/2 1.2，使得原先为这些系统开发的应用程序都可以在 Windows 2000 下运行。

（2）易移植性　Windows 2000 通过硬件抽象层（HAL）将内核、设备驱动程序以及执行体同硬件分隔开来，使它们可以适应多种平台。从而提高了易移植性。

（3）可靠性　Windows 2000 能主动地保护自身免受异常或外部有意或无意破坏的影响，并且对软件和硬件的错误做出可预测的响应。它的文件系统能自动从各种系统故障中恢复。

（4）高性能　为获得高性能并进而得到系统的灵活性，Windows 2000 在系统设计中采用了一些好的算法和**数据结构**以及先进的通信机制，如本地过程调用（LPC）。

（5）国际化　Windows 2000 基于 UNICODE，通过提供 NLS API，来支持不同地区的本地化使用。

主要特点

Windows 是系列产品，在它发展过程中的每一个新版本都有突出的新功能和新特点。它们已经对用户的工作方式和应用程序的开发产生了巨大影响，其影响力主要来源于贯穿整个产品系列的下列特点。

（1）多任务的图形化用户界面 Windows 系统从一开始就摆脱了字符形式的操作界面，为每个运行的程序提供了一个独立的窗口，窗口的大小、位置、显示方式均可由用户控制，在窗口内分层次合理地组织了标题条、控制选单框以及各种按钮，除需要输入正文参数外，仅用**鼠标器**就可以方便地进行操作，执行各种功能。Windows 系统支持多任务，集中管理对应于每个任务的窗口，用鼠标器很容易在各窗口之间切换，实现多任务件的切换。系统还支持动态数据交换，建立了任务间的数据联系。Windows 系统利用各种图示化手段，结合强大的联机帮助和提示机制，使得系统易学易用。

（2）事件驱动的程序运行方式 Windows 支持基于消息循环的程序运行方式，使应用程序也采用类似操作系统的运行方式，而消息产生于用户环境引发的事件（如鼠标器或**键盘**动作）。与应用程序传统的序列驱动方式相比，事件驱动方式有较大的灵活性，对用户交互操作较多的应用程序有明显的优点。

（3）标准的应用程序界面 Windows 系统为应用程序开发人员提供了功能强大的应用程序开发接口（API）。通过调用应用程序接口，开发者很容易创建 Windows 图形界面的各种元素，如窗口、选单、滚动条、对话框以及各种工具条等。其结果是应用程序在提供各自不同的功能时采用了风格一致的界面，也就是与 Windows 系统界面风格一致的界面。这不仅简化了应用程序开发，更重要的是大大简化了学习使用不同应用程序的过程。

Windows 系统还为应用程序开发提供了图形设备接口（GDI），实现了与设备无关的图形输出，使得应用程序能够以一致的方式调用同类设备。

（4）不断增强的功能 每一种新版本的 Windows 都带来许多新功能，其目标始终是充分发挥不断增强的硬件能力以及尽可能更易于使用。突出的例子包括对内存的管理以及从 16 位升为 32 位，Windows 系统早已突破 DOS 对内存地址空间的限制。在 Windows 的不断发展过程中，系统逐渐集成了许多原先的工具软件，甚至某些应用软件的功能，在支持网络、多媒体、安全性等方面有了很大发展。同时保持了良好的兼容性，并不断提高用户界面的友善性。

发展趋势

从总体上看，Windows 系统今后发展主要趋势是功能更强大，安全性更高，使用更方便。

目前 Microsoft 公司正致力于 .Net 计划，试图通过使用分布式计算模型（如 DCOM）和基于开放标准（如 XML），允许用户应用程序通过 Internet 进行**数据通信**和资源共享。这一庞大的“无处不在的计算”计划，既是基于操作系统已有的成果，又对操作系统提出了更高的要求。Windows 的未来之路无论是继续其版本延续，还是以全新的面貌出现，都将服务于 .Net 计划。

参考文献

1. 尤晋元，史美林，等. Windows 操作系统原理. 北京：机械工业出版社，2001

2. Silberschatz A，Galvin P B，Gagne G. Operating system concepts. 6th ed. John Wiley & Sons，Inc.，2002

3. Solomon D A, Russinovich M E. Inside Microsoft Windows 2000. 3rd ed. Microsoft Press, 2000

（陈道蓄　茅兵）

WS-BPEL

WS-BPEL（Web service business process execution language，Web 服务业务过程执行语言）　非营利性国际技术组织 OASIS（Organization for Advancement of Structured Information Standard）开发的用于描述业务过程的语言。如其名字所言的，设计者不仅希望该语言成为一个支持复杂业务流程的建模和设计语言，还希望它能在执行引擎支持下成为一种可以实际执行的系统开发语言。其设计考虑了一般过程和工作流的特点，特别考虑了设计和实现 Web 服务的需要。

WS-BPEL 的设计从 IBM 的 Web 服务工作流语言 WSFL 和微软的流程语言 XLANG 汲取了很多营养，其第一个版本称为 BPEL4WS（business process execution language for Web service），于 2002 年 7 月发布，修改完善后的 1.1 版于 2003 年 5 月发布。进一步修改完善并更名为 WS-BPEL 后的 2.0 版于 2007 年 4 月发布，作为本项工作的最后版本。

WS-BPEL 语言提供了许多常见的顺序与并发控制结构，基于 WSDL（Web service description language）规范的 Web 服务调用与信息接受操作，用于支持动作之间复杂约束关系的链接机制（link），较常规的数据处理机制如变量、表达式和赋值以及为支持开发复杂业务过程而提供的作用域（scope）结构，等等。这些基本元素和控制结构可以支持开发者较为方便地描述基于已有 Web 服务构造增值的 Web 服务系统（和其他业务流程系统）。

WS-BPEL 的设计中还特别提出了支持长时间运行事务（long-running transaction，LRT）的概念。这里的“事务”就是“事务处理”中的事务概念，其原子性语义要求：一个事务或者全部完成或者无影响（all or no）。LRT 是事务概念的发展，因为在实际应用中可能有一些持续时间很长的业务，在其工作过程中，特别是在 Web 环境里，环境状态不可能被锁定，环境变化也不可能完全控制。在这种情况下，WS-BPEL 提出了一种称为补偿（compensation）的概念，使设计师可以基于补偿处理和错误处理等机制描述已经部分进行的工作的撤销、调整或转换处理。此外 WS-BPEL 还提供了事件处理等许多很有价值的设计元素。目前一些公司已经发布了支持 WS-BPEL 的执行引擎，人们已经基于 WS-BPEL 开发了许多实际的 Web 应用。

参考文献

Web Services Business Process Execution Language, version 2.0, 11 April 2007, http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html

（裘宗燕）

XCY yuyan

XCY 语言（XCY language）　一种系统程序设计语言。由我国南京大学徐家福、中国科学院计算技术研究所仲萃豪、北京大学杨芙清等于 1978 年设计。XCY 语言规模适度，简明易用，主要用于书写系统程序。

XCY 语言是在开发 PASCAL、美国 DOD 红色语言及绿色语言等基础上，兼顾书写系统程序的需要与简明易用两方面而设计的。XCY 语言提供的程序结构成分除通常子程序外，还有模块和路径。

模块将逻辑上相关的对象（公用量、子程序和操作等）封装在一起，通过模块说明引入。模块说明含一组说明、移出表和移入表。模块说明不可执行，只有调用外部可见的子程序才能使用其中的数据和操作。XCY 模块可以嵌套，可以分别编译。模块可以处于三种不同的工作方式：管态、用户态、封锁中断。针对书写系统程序的要求，模块有三种：管程模块：管理调度实资源；类程模块：控制作业路径专用的虚资源；一般模块：满足一般用户算法要求。

路径定义类似于子程序：

〈路径〉∷ = path〈路径名〉〈形参部分〉〈路径体〉end〈路径名〉

〈路径体〉∷ =〈说明表〉〈语句表〉

但可以并发执行。

XCY 语言提供的数据类型包括以下 3 种：

（1）标准简单类型：整型、布尔型、字符型；

（2）非标准简单类型：字符串型、字位串型、子域型；

（3）构造类型：记录、数组、联合。

XCY 语言中与机器有关的成分主要是用于指定数

据对象的实际存储位置,包括变量、模块位置信息以及记录中各个域的位置信息(包含这种信息的记录定义为 md 记录)。

DJS 200 系列计算机上的 RT 操作系统完全用 XCY 语言书写。XCY 语言还被用于书写不同的编译程序。XCY 本身的编译程序也是用 XCY 语言书写的。

参考文献

徐家福,仲萃豪,杨芙清. 系统程序设计语言 XCY 的设计. 北京: 计算机学报,1981,4(1):1-12

(陈道蓄)

XML shuju guanli

XML 数据管理(XML data management)

对 XML 数据进行存储、索引、查询、发布、集成等处理的技术。

可扩展标记语言 XML(extensible markup language)是以文档管理为基础,由标准通用语言 SGML(standard generalized markup language)派生出来。目前,XML 已经成为 Internet 环境中数据表示和交换的标准。XML 文档中基本的结构是元素,一个元素由一对开始和结束标签及它们之间的文本所构成。XML 文档要求元素正确嵌套,并且只能存在一个根元素。XML 元素中还允许包括属性的定义,属性和其取值在同一元素中只出现一次。XML 文档类型定义是 XML 文档的可选部分。XML 文档类型定义中通过正则表达式限制元素中合法的子元素和属性。目前,DTD(document type defination)和 XML Schema 是描述 XML 文档类型的两种标准。

XML 数据模型由表示 XML 文档的节点标记树、节点标记树上的操作和语义约束组成。XML 节点标记树中包括不同类型的节点。其中,文档节点是树的根节点,XML 文档的根元素作为该文档节点的子节点;元素节点对应 XML 文档中的每个元素;子元素节点的排列顺序按照 XML 文档中对应标签的出现次序;属性节点对应元素相关的属性值,元素节点是它的每个属性节点的父节点;命名空间节点描述元素的命名空间字符串;节点标记树的操作主要包括树中子树的定位以及树和树之间的转换。XML 元素中的 ID/IDref 属性提供了一定程度的语义约束的支持。

XML 数据管理中 XML 存储是后续操作的基础。存储 XML 数据的一种可选方法是将 XML 数据保存到关系数据库中。由于 XML 文档中元素是嵌套的,并且可能具有重复的子元素,这些差异使得不能通过简单、直接的映射来存储 XML 数据。利用**关系数据库**实现 XML 存储的具体实现方法有将 XML 作为字符串来存储、利用表结构存储 XML 数据树中的节点信息和边信息、或者将 XML 数据项分别存储到不同关系中。利用关系数据库来存储 XML 数据,还要求系统将 XML 查询转化成关系数据库查询。由于数据模型不同,单一的 XML 查询可能转换成复杂的关系查询。存储 XML 数据的另一种方法是采取非关系的方式来存储 XML 数据,例如设计纯 XML 数据库来保存 XML 数据。

XML 数据查询能够从 XML 文档中提取特定信息,完成不同结构的 XML 文档之间互相转换。目前, XML 数据查询主要有三种语言:XPath 是基于路径表达式的语言,在 XML 节点标记树中定位相关的子树,事实上也是 XQuery 的基础成分;XSLT(extensible stylesheet language transformation)是一种转换语言。可以利用 XSLT 定义一系列具有匹配功能和选择功能的模板,将 XML 数据转换到 HTML 语言或者其他和数据显示相关的语言;XQuery 是 XPath 的超集,也是万维网联盟 W3C 推荐的 XML 数据查询的标准语言,包含 For、Let、Where、Order by、Return 等子句,支持用户转换 XML 数据树,并日益得到厂家的支持。

XML 数据管理的实现方式可以采用纯 XML 数据库系统的方式,纯 XML 数据库基于 XML 节点树模型,能够较自然地支持 XML 数据的管理。但是,纯 XML 数据库需要解决传统关系数据库管理所面临的各项问题,包括查询优化、并发、事务、索引等问题。目前,很多商业关系数据库通过扩展的**关系代数**来支持 XML 数据的管理。扩展的**关系代数**不仅仅包含传统的关系数据操作,而且支持 XML 数据特定的投影、选择、连接等运算。传统的查询优化机制也要加以扩展来满足新的 XML 数据操作的要求。通过关系数据库查询引擎的内部扩展, XML 数据管理能够更加有效地利用现有关系数据库成熟的查询技术。

参考文献

1. Shanmugasundaram J, Tufte K, Zhang C, et al. Relational databases for querying XML documents: limitations and opportunities. VLDB, 1999: 302-314

2. Liu L, Ozsu M T: Encyclopedia of database systems. Springer, 2009

(高军 杨冬青)

XYZ/E yuyanzu

XYZ/E 语言族（XYZ/E language family）

一种系列化的时序逻辑语言族，其中各子语言分别表示不同的程序设计方式或程序范型的语言。XYZ/E由中国科学院计算技术研究所唐稚松于20世纪70年代设计，它的最基本的特征是以一种统一的框架既能表示适应诺依曼体系的状态转换机制的命令式语言，又能表示适应逻辑推理特征的直言式公式语言。

XYZ/E 有三种控制结构：一种是直接表示状态转换的命令形式，具有这种控制结构的子语言称XYZ/BE（即 Basic XYZ/E）；另一种则是结构化高级语言的语句形式，具有这种控制结构的子语言称为 XYZ/SE（即 Structured rule form XYZ/E）；第三种控制结构则是产生式规则的形式，具有这种控制结构的子语言称为 XYZ/PE（即 Production rule form XYZ/E）。

XYZ/E 中也包括了表示各种并发性或不确定性、不同通信方式、不同类型的可重用模块的机制。故在一统一的程序中可包含所有这些机制及相应的各种程序设计方式。它同时还能包含表示多种可视图形程序的语义，而且这些图形与相应的 XYZ/E 程序可相互自动生成。由可重用模块（过程、进程、包块）与并发通信机制结合而成的程序，由于其结合方式不同，可以构成差异很大的总体结构，其中有些情况是互不相容的。可以区分为三种类型：①非分布式环境下基于对象的程序；②非分布式环境下面向对象的程序；③分布式程序。

XYZ 系统是将时序逻辑与软件工程有机结合、基于 XYZ/E 语言的计算机辅助软件工程（CASE）环境。故它构成一正交的二维体系。一维是基于 Manna-Pnueli 线性时间时序逻辑语言族 XYZ/E，另一维是 CASE 工具集，包括五组工具（交互式验证工具与自动生成工具、记录历史的逐步求精与原型速成工具、结构化设计的可视图形工具、语言转换工具、软件管理工具），均以 XYZ/E 表示其语义界面，它们既可独立使用又可根据其输入输出界面语义一致性相互连接组成更复杂的工具。

（柳军飞　赵琛）

X.25 fenzu jiaohuan wangluo

X.25 分组交换网络（X.25 packet switching network）　在国际电信联盟 ITU-T 建议标准系列中，X 系列是关于数据通信网的若干建议，其中 X.25 建议用于分组交换网中连接**数据终端设备**（DTE）和**数据电路设备**（DCE）之间的接口协议标准。该标准 1976 年起正式成为国际标准，并在历次国际标准会议上补充修订。

在**开放系统互连 OSI 参考模型**中，X.25 建议定义了七层网络体系结构中下三层协议，即物理层、数据链路层和网络层的协议，如图 1 所示。

物理层主要定义 DTE 和 DCE 之间接口的机械的、电气的、功能的和协议的特性，物理层传输比特流信息，所采用的标准有 X.21/X.21bis 建议和 V.24/RS232C 建议，V.24 与 RS232C 建议是兼容的。

数据链路层协议也叫帧级协议，在数据链路层是以帧结构为基本单元传送信息。在物理层提供的比特流传输基础上，在数据终端设备（DTE）和数据电路设备（DCE）之间进行透明、可靠的帧级数据传输。数据链路层的标准包含高级数据链路控制（HDLC）协议的子集 LAP（链路访问协议）和 LAP-B

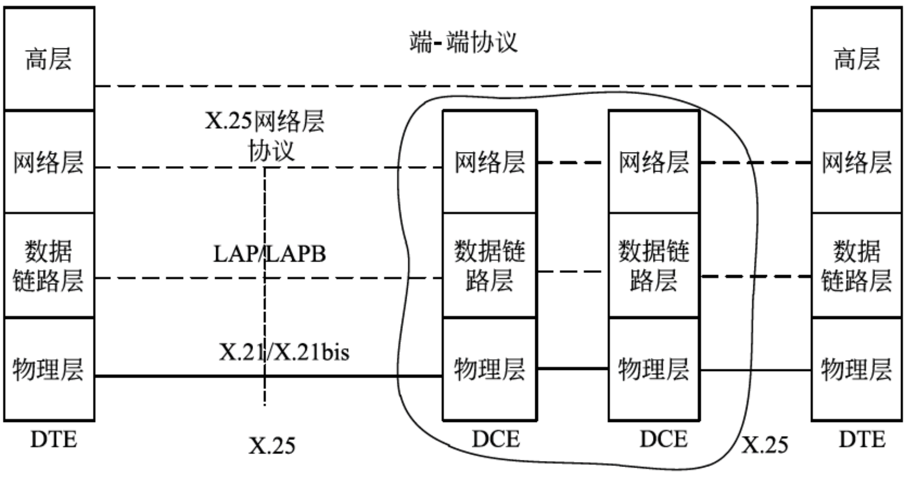


图 1　X.25 协议体系结构

（平衡型链路访问协议）。

网络层是分组信息传输层，提供虚电路（VC）服务（虚电路建立、维护和释放）、寻址与路由选择、流量控制、顺序检验和差错控制等，以保证网络终端用户（DTE 与 DTE）间透明可靠的数据传输。

X.25 分组交换网

分组交换是与电路交换不同数据通信交换方式，在分组交换网络中，由于采用电路的动态统计复用原理，即在一条物理线路上同时开放多条虚电路，为多个用户使用，从而提高了网络效率，以 X.25 建议为基础的分组交换网称为 X.25 分组交换网。可以满足不同速率、不同型号终端与计算机、计算机与计算机间以及局域网间的通信，实现数据库资源共享。分组交换网是一种基础的数据通信网络，在其网络平台上可以架构各种增值业务，如电子信箱、电子数据交换（EDI）、传真、存储转发等。

X.25 分组交换网在 20 世纪七八十年代全球应用较多，我国于 1993 年正式建成投产 X.25 分组交换网（CHINAPAC），第一次在国内开放数据通信业务，满足了当时计算机和数据终端联网需求，随着帧中继、ATM 特别是 IP 技术的发展，X.25 分组交换网逐渐被新技术替代，CHINAPAC 也于 2005 年退出历史舞台。

参考文献

Tanenbaum A S. Computer networks. 3rd ed. Englewood Cliffs, NJ: Prentice Hall, 1996

（马妍　马严）

X3D tuxing biaozhun

X3D 图形标准（X3D Graphics Standard）

包含三维图形内容或集成多媒体信息的网络交换文件的格式标准（ISO/IEC 19775）。

X3D 的应用领域包括网络三维数据的实时渲染、科学和医疗可视化（参见**可视化**）、虚拟现实、模拟训练、多媒体、娱乐和教育等。

X3D 标准是原网络三维图形标准（ISO/IEC 14772）的虚拟现实建模语言（VRML）的扩展，具有更先进的应用程序接口和更严格的数据一致性，支持更多的数据编码格式以及部件化（componentized）体系结构。X3D 在原 VRML 节点的基础上，定义了一套基于可扩展标记语言（extensible markup language，XML）的描述，使得 X3D 文件更适于分布式存储和网络传输，并能够更容易地实现异构数据互操作，更适于作为通用的三维图形及集成多媒体信息的交换格式。

X3D 的主要特征是：

（1）内建了对交互和对象行为的支持，现实世界中各种事物之间的相互联系可以通过标准中定义的各种传感器和事件传输路由来进行描述，还可以通过插入脚本（script）来描述更为复杂的事物行为方式，用户可以通过各种方式与用 X3D 描述的场景进行实时互动，由于各种事物的行为方式不是通过硬编码而是通过 X3D 本身来描述的，因此能够动态地对其进行修改。

（2）采用了独立于数据编码的部件化可扩展构架，定义了一个精巧而易于实现的内核，在此基础上通过扩展各种部件来满足不同应用的要求。在核心部件之外，X3D 目前已定义了 20 多个基本部件，主要包括：描述三维场景中各种几何形体的形状部件、三维几何部件、二维几何部件、文本部件、非均匀有理 B 样条部件和组合部件；描述场景中各种真实感效果的光照部件、纹理部件、环境效果部件和绘制部件；描述动画和基本用户交互的计时器部件、插值器部件、漫游部件、点选设备和按键输入部件、环境传感器部件、脚本部件和事件应用程序部件；描述特定应用领域对象的地理空间信息部件、**分布式交互仿真**（distributed interactive simulation，DIS）部件、人体动画部件；定义网络传输的网络部件；定义多媒体集成的声音部件等。

（3）针对不同的应用可以选择合适的部件配置（profile），实现对相关部件的特定程度的支持。X3D 给出了 5 种配置方式，分别是交换（interchange）配置、交互（interactive）配置、MPEG-4 交互配置、沉浸感配置和完全（full）配置，不同的配置反映了对 X3D 部件不同程度的支持。另外，X3D 是一个开放的标准，开发者可以针对领域中的特定情况，在现有基本部件基础上扩展定义自己的部件。

参考文献

ISO/IEC 19775: 2008. Information Technology-Computer graphics-Image processing and environmental representation, 2008

（王青）

ZigBee

ZigBee（ZigBee）　基于 IEEE 802.15.4 标准的低功耗个域网协议。根据这个协议规定的技术是一种短距离、低功耗的无线通信技术。这一名称来源于蜜蜂的八字舞，由于蜜蜂（bee）是靠飞翔和“嗡嗡”（zig）地抖动翅膀的“舞蹈”来与同伴传递花粉所在方位信息，也就是说蜜蜂依靠这样的方式构成了群

体中的通信网络。其特点是近距离、低复杂度、自组织、低功耗、高数据速率。主要适合用于自动控制和远程控制领域,可以嵌入各种设备。　　（周正）

Z yuyan

Z 语言(Z language)　一种以状态机为模型的形式规约语言。最早由法国人 J. R. Abrial 提出,由 C. A. R. Hoare(英国牛津大学)领导的程序设计研究小组发展而成。

Z 的基本单位是模式。一个模式用一右边开口的矩形框起,中间一道横线将它分成说明部分和谓词部分。说明部分定义一些状态或模式变量;谓词部分是一般的谓词公式,它给出了变量间的限定关系。模式可定义系统的状态空间,初始状态和状态变换。如下面的模式定义了公寓出租系统的状态空间:

RentApart
apartments: **P** APARTMENT
customers: **P** PERSON
booked: APARTMENT→TIME→PERSON

dom booked ⊆ apartments
∀(ap:APARTMENT | ap ∈ apartments.
ran booked ap ⊆ customers)

其中 RentApart 是模式名。apartments 是一个公司所有的公寓;customers 是该公司的客户集。booked 是一个部分函数,它指明某一公寓在特定时间里租给了某个客户。谓词部分指出,booked 的定义域(公寓)必须是该公司的财产,而租用人(booked 的值域)必须是该公司的客户。说明和谓词部分都可以有多行,行与行之间分别是"拼接(;)"关系和"与(∧)"关系。模式也可以线性地表示,如上面的模式可表示为:

RentApart ≡ [apartments: **P** APARTMENT; customers: **P** PERSON;
booked: APARTMENT → TIME →PERSON
| (dom booked ⊆ apartments) ∧
∀(ap ∈ APARTMENT | ap ∈ apartments ∧ ran booked ap
⊆ customers)
]

由此可见,一个模式定义了一个(子)集合。作为集合,模式可以进行诸如并(∨),交(∧),蕴含(⊇),幂集(P),大小(#)等集合运算,也可以移入已定义的模式。

模式既可以表示数据状态,也可以表示运算。运算通过状态变化来表示。当用来表示运算时,用 ΔS 表示模式 S 的状态改变,用 a′表示变量 a 改变后的状态值。如:

Booking
ΔRentApart
a?: APARTMENT
t?: TIME
p?: PERSON

a? ∈ apartments
p? ∈ customers
t? ∉ dom booked a?
booked′ = booked ∪ {a? →t? →p?}
apartments′ = apartments
customers′ = customers

表示,当一间公寓(a ?)某个时间(t ?)未租出,则将它租给客户(p ?)。这里,变量后面的"?"表示该变量是输入变量;若要表示输出,则在变量后面加"!"。

一个 Z 规约就是一组模式和用自然语言书写的对各模式的注释。模式之间通过模式运算和移入相关联。

Z 以其直观、简洁、接近功能分解模型吸引了众多的用户。利用谓词演算和模式演算的数学性质,可对 Z 规约进行形式分析。现已有各种 Z 规约辅助开发工具。针对面向对象的软件风范,Z 又有了 Object Z 和 Z++ 等扩充。

参考文献

1. Spivey J M. Introducting Z: a specification language and its formal semantics. New York: Cambridge University Press, 1988

2. Spivey J M. The Z notation: a reference manual. New York: Prentice Hall. Inc., 1989　　（伊波）

λ yansuan

λ 演算 (λ calculus)　一种研究函数的定义与求值的一般方法。它是一种等价于图灵机理论的计算模型。λ 演算既是函数式程序设计语言的数学模型,又是程序设计语言的指称语义的理论基础。λ 演算与直觉主义逻辑相结合为交互式证明编辑系

统等软件工具提供了逻辑框架。

第一个 λ 演算系统是由美国学者A. Church为解决数学基础问题提出的。A. Turing 和 J. C. Kleene 证明了图灵机递归函数与 λ 演算作为计算模型的等价性,导致了图灵-立奇论题。美国学者 J. B. Rosser,H. Curry 等都对 λ 演算的发展做过重要贡献。1969 年 D. Scott 发展了 λ 演算的模型论,并与 Strachey 合作给出了程序设计语言的指称语义。20 世纪 60 年代J. McCarthy 设计并实现了第一个以 λ 演算为模型的程序语言 Lisp。1978 年 J. Backus 倡导使用基于 λ 演算与组合逻辑的函数式程序设计语言,推动了程序语言的研究和发展。80 年代初 P. Martin-Löf将直觉主义逻辑思想引入计算机科学,使用带类型 λ 演算,给出了通过对程序规约进行证明,而直接构造程序的开发方法,进一步扩大了 λ 演算在程序理论中的应用,也推动了高阶带类型 λ 演算的研究。

λ 演算的主要理论包括它的语法范畴,归约理论,模型论以及带类型 λ 演算等几个部分。

λ 项是 λ 演算的基本对象。令 V 为变元组成的可数无穷集合,λ 项可以递归定义如下:

(1) 若 $x \in V$,则 x 是一个 λ 项。

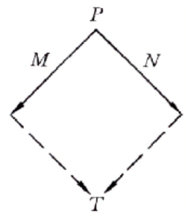
(2) 若 M,N 都是 λ 项,则 MN 是一 λ 项。

(3) 若 M 是 λ 项而 $x \in V$,则 $\lambda x.M$ 是一 λ 项。

例如 xx,$\lambda x.x$,$\lambda x.xy$,$(\lambda x.x)(\lambda x.x)$ 等都是 λ 项。形如 $\lambda x.M$ 的 λ 项称为一个抽象,x 称为约束变元,M 是 x 的作用域。它大体相当于一个程序设计语言的过程说明,其中 M 相当于过程体,x 相当于形参,λ 相当于过程说明标记。MN 称为一个作用,它相当于一个过程调用,N 相当于实参。不同的是在 λ 演算中 M 和 N 可以是任意 λ 项。

若变元 y 不在 M 中自由出现,则用 y 替换抽象 $\lambda x.M$ 中 M 的自由变元 x,记成 $\lambda y.[y/x]M$,称为对此 λ 项的一个 α-转换,如果 M 是由 N 经有限步 α-转换而得,则称 M 与 N 是同余的。这里$[y/x]M$ 表示用 y 替换在 M 中 x 的所有自由的出现。$(\lambda x.M)N$ 称为β-可约项,而相应的$[N/x]M$ 称为一个约减项。若一 λ 项不包含 β-可约项,则称此项为 β-典范式,将一个 λ 项中的β-可约项用它相应的约减项替换称为对此 λ 项的一次 β-归约。若 λ 项 N 是由项 M 经有限次β-归约及α-转换而得,则称 Mβ-归约到 N,记为 $M \to_\beta^* N$。如果存在 $M_0, M_1, \cdots, M_n$,使 $M_0 \equiv M$,$M_n \equiv N$,且对任何 $i<n$ 有 $M_i \to_\beta^* M_{i+1}$ 或 $M_{i+1} \to_\beta^* M_i$,则称 M 与 Nβ-相等,记为 $M=N$。可以证明:对任意 λ 项 F,都存在一 λ 项 M,使得 $FM=M$ 成立,这就是 λ 演算的不动点定理。

可以证明 λ 演算具有立奇-罗瑟性质。该性质说:如果 λ 项 P 可以β-归约到 M 和 N,则必存在一 λ 项 T,使 M 和 N 分别β-归约到 T。



这就是β-归约的立奇-罗瑟性质。

并非每个 λ 项都具有 β-典范式,例如$(\lambda x.xx)(\lambda x.xx)$是一个β-可约项,对它进行一次 β-归约后得到$[(\lambda x.xx)/x]xx$,再用 $\lambda x.xx$ 分别替换项 xx 中的两个 x,又得到$(\lambda x.xx)(\lambda x.xx)$,因此对这个 λ 项,β-归约可以无限长。1932 年 A. Church 证明判断任意一个 λ 项是否具有 β-典范式是一不可判定问题。

可以递归地将某些 λ 项赋予一个类型,每个类型可以看成某些 λ 项组成的集合,做法如下:把变元 x 赋予类型 σ,记为 $x{:}\ \sigma$;若 λ 项 M 的类型为 η,则 $\lambda x{:}\ \sigma.M$ 就是一带类型的 λ 抽象,它的类型为 $\sigma \to \eta$。若 M,N 都是带类型 λ 项,它们的类型分别为 $\sigma \to \eta$ 及 σ,则 MN 也是一带类型的 λ 项,它的类型为 η。对带类型的 λ 项可以定义相应的 α-转换,β-归约,并建立相应的归约理论,带类型的 λ 演算也具有不动点定理,立奇-罗瑟等性质。其中,最重要的结果是强典范化性质,即对任意的带类型λ 项,从它开始任意的归约序列都是有穷的,从而带类型λ 项是否有典范式问题是可判定的。这就是许多程序设计语言中变量、函数和过程都具有类型,并由编译程序进行类型检查的理论基础。

前面关于 λ 项、α-转换、β-归约都是形式定义的,因此 λ 演算可以定义成一个形式系统。能否找到一个数学理论作为 λ 演算形式系统的模型是 λ 演算模型论的主要研究问题。其难点是:有些 λ 项,如$(\lambda x.x)(\lambda x.x)$是形式上合理的β-归约项,它可以β-归约为$(\lambda x.x)$,但如果将 x 解释为某集合 σ 的元素,则 $\lambda x.x$ 就解释为定义域、值域均为 σ,即类型为 $\sigma \to \sigma$ 的函数。这个函数在集合论里只能被定义域 σ 中的元素作用,而不能被 $\sigma \to \sigma$ 中的元素作用,因此$(\lambda x.x)(\lambda x.x)$在经典集合论中就不能得到合理的解释。1969 年 D. Scott 建立了**论域理论**。其

基本对象是完全偏序集(简记为 cpo)和建立在完全偏序集上的连续函数。D. Scott 首先证明,如果 D_0 是一任意 cpo,则以 D_0 为定义域和值域,类型为 $D_0 \to D_0$ 的全体连续函数也构成一个 cpo,记为 $D_1 \equiv [D_0 \to D_0]$,且存在 D_0 到 $[D_0 \to D_0]$ 的一一对应关系。顺此,令 $D_{n+1} \equiv [D_n \to D_n]$ 可以构造下述完全偏序集序列

$$D_0, D_1, \cdots, D_n, D_{n+1}, \cdots$$

D. Scott 进一步证明上述序列收敛到一完全偏序集 D_∞,且满足 $D_\infty = [D_\infty \to D_\infty]$。这样 $(\lambda x.\, x)(\lambda x.\, x)$ 在 D_∞ 中就成为合法的连续函数了。使用 D. Scott 的模型论思想可以将程序设计语言中诸如 *while* **true** *do* *skip* *end* 这类死循环语句,解释为相应完全偏序集上的连续函数,从而为指称语义学奠定了理论基础。

参考文献

1. Barendregt H P. The lambda calcalus, its syntax and semantics. North Holland, 1984

2. Hindley J R, Seldin J. Introduction to combinators and λ - calculus. Cambrigde University Press, 1986

(李未　王驹)

π yansuan

π 演算(π-calculus)　基于**通信系统演算**(CCS)的可描述通信拓扑结构动态变化的分布式通信系统(例如移动电话系统)的传名演算。又称移动进程演算。π 演算由 R. Milner, J. Parrow 和 D. Walker 三人提出。该演算允许进程之间传送和接收通道名,并且可以引入和输出局部名。因此,π 演算不仅非常简洁,而且表达能力很强。

π 演算的基本实体是通道名,通道名一方面可用于标识通道,另一方面可作为通道传送的消息。令 $\mathcal{N}$ 是通道名的可数无穷集,其元素用 $a, b, x, y, \cdots$ 表示,令 t 表示进程,π 演算的语法可由以下 BNF 给出:

$$t ::= 0 \mid \alpha \cdot t \mid t + t \mid [x = y]t \mid t \mid t \mid (x)t \mid A(y_1, \cdots, y_n)$$

$$\alpha ::= \tau \mid a(x) \mid \bar{a}x$$

π 演算的进程构造算子大部分来源于 CCS:0 为不活跃进程,$\alpha \cdot t$ 为动作前缀,+ 为非确定选择,$[x=y]t$ 为等名测试,| 为并发合成,$(x)t$ 为限名,表示 x 是 t 的私有名。每个进程常量 A 都有一个定义方程 $A(x_1, \cdots, x_n) \Leftarrow t$。

在 π 演算中有三类基本动作:τ 表示内部通信动作,$a(x)$ 表示输入动作,$\bar{a}x$ 表示输出动作。当 $a \neq x$ 时,可将 $(x)\bar{a}x \cdot t$ 缩写为 $\bar{a}(x) \cdot t$,导出动作 $\bar{a}(x)$ 称为约束输出动作。在 $a(x) \cdot t$, $(x)t$ 和 $\bar{a}(x) \cdot t$ 中,x 为具有辖域 t 的约束名。由此引出进程项 t 的约束名字集 $bn(t)$ 和自由名字集 $fn(t)$。

与传值 CCS 不同的是,由于 π 演算不区分数据变元和通道名,故限名算子 (x) 不仅能限制动作的主体——通道,而且能限制动作的客体——被传送的名字,如 $(x)\bar{a}x \cdot t$。因其私有性,被限制的私有名不能作为通道与外部通信。但能通过其他的通道向外界输出,进而将其辖域扩展到接收该私有名的进程。这一点是 π 演算与传值 CCS 的主要差异,也是导致 π 演算表达能力丰富和语义复杂的根源。因此在 π 演算中有两类含义不同的约束名:一类是形如 $(x)t$ 的私有名 x,不能被实例化;另一类是形如 $a(y) \cdot t$ 的输入参数名 y, y 只是一个占位子,可被实例化。这两类约束名都可进行 α 换名。

π 演算的迁移语义如下所示:

pre $$\frac{-}{\alpha \cdot t \xrightarrow{\alpha} t}$$

sum $$\frac{t \xrightarrow{\alpha} t'}{t + u \xrightarrow{\alpha} t'}$$

match $$\frac{t \xrightarrow{\alpha} t'}{[x = x]t \xrightarrow{\alpha} t'}$$

par $$\frac{t \xrightarrow{\alpha} t'}{t \mid u \xrightarrow{\alpha} t' \mid u} \quad bn(\alpha) \cap fn(u) = 0$$

com $$\frac{t \xrightarrow{a(x)} t', \ u \xrightarrow{\bar{a}y} u'}{t \mid u \xrightarrow{\tau} t'[y/x] \mid u'}$$

res $$\frac{t \xrightarrow{\alpha} t'}{(x)t \xrightarrow{\alpha} (x)t'} \quad x \notin n(\alpha)$$

open $$\frac{t \xrightarrow{\bar{a}x} t'}{(x)t \xrightarrow{\bar{a}(x)} t'} \quad x \neq a$$

close $$\frac{t \xrightarrow{a(x)} t', \ u \xrightarrow{\bar{a}(x)} u'}{t \mid u \xrightarrow{\tau} (x)(t' \mid u')}$$

id $$\frac{t[\tilde{y}/\tilde{x}] \xrightarrow{\alpha} t'}{A(\tilde{y}) \xrightarrow{\alpha} t'} \quad A(\tilde{x}) \Leftarrow t$$

由于在 π 演算中抹杀了常量名和变量名、数据名和通道名之间的区别,为 π 演算引入基于互模拟

概念的行为等价语义理论时将出现新的问题。直接根据强互模拟基本定义得到的关系只是一个等价关系,而不是同余关系,它不能被名字替换所保持,因而也不能被输入动作所保持。

将类似于传值 CCS 的这种基本的互模拟关系称为基互模拟。很自然地,π 演算的互模拟同余关系(简称为互模拟同余)“~”可定义为能被所有名字替换保持的基互模拟关系,即,$t \sim u$ 当且仅当对任意名字替换 σ,均有 $t\sigma \sim u\sigma$。此时所有的自由名都看成变量。互模拟关系的这种二步定义法是 π 演算的一个有趣的现象和重要的特色。当然,与传值 CCS 类似,根据对输入动作的参数名的不同实例化策略,π 演算进程之间的基互模拟等价和相应的互模拟同余均有迟、早之分。

在 π 演算中,名字用来引用对象,一个进程所具有的自由名代表了该进程当前所拥有的关于其他进程的知识或与其他进程的联系。所有其他对象都对等地视为进程,通过原子事件进行交互。利用所谓“分子动作”技术,所有复杂数据类型以及基本函数,都可通过编码定义为进程,而不作为基本对象。π 演算进程间所能传送的对象只能是通道名,而不能为进程,即只能用传递引用代替传递进程。虽然 π 演算是一阶的,但利用通道名作为指向进程的指针,容易将高阶进程翻译到 π 演算中去。

π 演算是对应用的广度(理论的一般性)和理论的完美同时追求的结果,通过较高抽象级来获得二者的统一。就概念简洁性和一般性而言,π 演算类似于作为函数计算基础的 λ 演算,因而可作为研究并发通信系统的基本工具。

参考文献

1. Milner R, Parrow J, Walker D. A calculus of mobile processes, Parts Ⅰ and Ⅱ. Journal of Information and Computation, 1992,100:1-77

2. Bergstra J A, Ponse A, Smolka S A, eds. Handbook of process algebra. Elsevier, 2001

(李舟军)

ω - youxian zidongji

ω - 有限自动机 (ω-finite state automaton)

一种在无限串上运行的有限状态自动机,是一种 ω - 语言的识别模型。主要研究 ω - 的各种识别方式以及在通常的五种识别条件下,识别的 ω - 语言族之间的关系。特别,通过其中一种条件(即所谓 C_5)下识别的 ω - 语言定义了 ω - 正则语言,这是一种使 ω - 自动机识别能力最强的识别方式。ω - 自动机理论的核心课题之一,是对 ω - 正则语言的研究,包括对 ω - 正则语言的描述及其性质的研究。

ω - 自动机最早在文献中出现的是 J. R. Buchi (1960)利用工作在无限序列上的有限自动机获得关于受限二阶逻辑理论的一个判定过程。自此以后一些研究 ω - 自动机的各种形式体系的论文陆续出现,其中 J. R. Buchi (1965, 1969), C. C. Elgot 和 M. O. Rabin (1966,1969)等人的论文均受到这些模型与二阶逻辑理论之间的密切关心的启发,因此重点放在判定问题。D. E. Muller (1963)利用确定的 ω - 有限自动机研究异步开关理论中的某些问题。R. MeNaughton (1966)首先发展了被 ω - 有限自动机识别的 ω - 语言的理论,即所谓的 ω - 正则语言的理论。

ω - 有限自动机研究的内容包括 ω - 有限自动机的定义,五种识别条件,ω - 正则语言的概念,对 ω - 正则语言的描述以及与五种识别模型相应的五个 ω - 语言族之间的关系。

ω - 串与 ω - 语言　设 Σ 是有限字母表,由 Σ 中的字母组成的无限序列,称为 Σ 上的 ω - 串。用 Σ^ω 表示 Σ 上的所有 ω - 串的集合。Σ^ω 的任意子集称为 Σ 上的 ω - 语言。

ω - 有限自动机　一个五元组 $M = (K, \Sigma, \delta, q_0, F)$,其中 K 为状态有限集,Σ 为输入字母表,$\delta: K \times \Sigma \to 2^K$,$q_0 (\in K)$ 为初始状态,$F (\subseteq 2^K)$ 为指定状态集族。如果 $\delta: K \times \Sigma \to K$,则 M 是确定的ω - 有限自动机。

设 $\sigma = a_1 a_2 \cdots a_n \cdots, a_i \in \Sigma, i = 1, 2, \cdots$。状态序列 $r = \{q_i\}$,称为 M 在 σ 上的一个运行,当且仅当 $q_i \in \delta(q_{i-1}, a_i), i = 1, 2, \cdots$。一个运行确定一个映射 f_r: $N \to K, f_r(i) = q_{i-1}, i = 1, 2, \cdots$。令 $I(r) = \{q \in K | \operatorname{card}(f_r^{-1}(q)) \geqslant \omega\}$,$O(r) = \{q \in K | f_r^{-1}(q) \neq \varnothing\}$。

ω - 有限自动机的识别条件　包括 C_1, C_2, C_3, C_4 与 C_5 五个条件。ω - 有限自动机 M 在 C_i 条件下识别 ω - 串 σ,当且仅当存在 M 在 σ 上的一个运行 r,使满足 $C_i, i = 1, 2, 3, 4, 5$。其中

C_1: 存在 $H \in F$,使 $I(r) \cap H \neq \varnothing$

C_2: 存在 $H \in F$,使 $I(r) \subseteq H$

C_3: 存在 $H \in F$,使 $O(r) \cap H \neq \varnothing$

C_4: 存在 $H \in F$,使 $O(r) \subseteq H$

C_5: 存在 $H \in F$,使 $I(r) = H$

设 $M = (K, \Sigma, \delta, q_0, F)$ 是一 ω - 有限自动机,称集合

$$T_i(M) = \{\sigma \in \Sigma^\omega \mid \text{存在} M \text{在} \sigma \text{上的一个运行} r, \text{使满足} C_i\}$$

为 M 在 C_i 条件下识别的 ω - 语言，$i=1,2,3,4,5$。

ω - 正则语言　Σ 上的一个 ω - 语言 L，如果存在一个 ω - 有限自动机 M，使 $T_5(M)=L$，则称 L 为一个ω - 正则语言。R. Hosseley 证明了在上述五种识别模型中，在 C_5 条件下 ω - 有限自动机的识别能力最强。因此，在 C_5 条件下的识别常常简单地说成“识别”，而用 $T(M)$ 表示被 ω - 有限自动机 M 识别的 ω - 语言，即

$$T(M) = \{\sigma \in \Sigma^\omega \mid \text{存在} M \text{在} \sigma \text{上的一个运行} r, \text{使} I(r) \in F\}。$$

具有单指定集的ω - 有限自动机　一种指定集族为单一集合的 ω - 有限自动机，称为单指定集的 ω - 有限自动机。容易证明，对于 C_1，C_3，单指定集的 ω - 有限自动机与一般的 ω - 有限自动机等价。

ω - 克林尼闭包　一种描述 ω - 正则语言特征的工具。令 $\mathscr{L}$ 表示正则集族，ω - 语言族

$$\omega\text{-}KC(\mathscr{L}) = \{L \subseteq \Sigma^\omega \mid L = \bigcup_{i=1}^{K} U_i V_i^\omega,\ U_i,\ V_i \in \mathscr{L},\ i=1,\cdots,K, K=1,2,\cdots\}$$

称为 $\mathscr{L}$ 的 ω - 克林尼闭包。

ω - 正则语言的特征定理　对 Σ 上的 ω - 语言 L，下列四个结论等价：

（1）L 属于 ω - $KC(\mathscr{L})$；

（2）存在一个识别 L 的 ω - 有限自动机；

（3）存在一个在 C_1 条件下识别 L 的单指定集的 ω - 有限自动机；

（4）存在一个识别 L 的确定的 ω - 有限自动机。

参考文献

Cohen R S, Gold A Y. Theory of ω - languages. 1: characterization of ω - context - free languages. Journal of Computer and System Sciences, 1977, 15(2): 169-184

（苏锦祥）

Σ(jidiao) daishu

Σ(基调)代数 (Σ(signature) algebra)　一种非齐性代数，由 G. Birkhoff 等于 1970 年作为对齐性代数的推广而引进。在非齐性代数中，元素集被分成几个互不相交的子集。每个代数运算均以特定的子集为其定义域和值域。描述这种结构的语法称为基调。

基调和 Σ 代数　令 $S=\{s_i \mid i \in I\}$ 为一有限集，其中 I 是一个有限下标集，每个 s_i 称为一个类子(可以看作 Σ 代数中元素的类型)。$O=\{O_j \mid j \in J\}$ 为另一有限集，J 也是有限下标集，每个 O_j 称为一个运算。一个 k 目运算 $O_j(k \geqslant 0)$ 可表为

$$O_j: s_1 \times s_2 \times \cdots \times s_k \to s_{k+1} \tag{1}$$

其中 $s_1,\cdots,s_k,s_{k+1} \in S$。对偶 $\Sigma=\langle S,O\rangle$ 称为一个基调。

给定基调 $\Sigma_0=\langle S,O\rangle$。假定有一组集合 $A=\{A_i \mid i \in I\}$ 和一组函数 $G=\{f_j \mid j \in J\}$，使得诸类子 s_i 和诸集合 A_i 之间有一一对应关系，诸运算 O_j 和诸函数 f_j 之间也有一一对应关系，且 $\forall i \neq j, A_i \cap A_j = \varnothing$。若函数 f_j 与式(1)的 O_j 相对应，则 $f_j: A_1 \times A_2 \times \cdots A_k \to A_{k+1}$。满足这些条件的对偶 $\langle A,G\rangle$ 称为一个以 Σ_0 为基调的 Σ 代数(或基调代数)，A 是它的载体集。

例如，为了定义基调“整数堆”，可设立 3 种类子：$s_1=\mathrm{int}$，$s_2=\mathrm{bag}$，$s_3=\mathrm{bool}$，和 9 个运算：

empty: ⟶bag
zero: ⟶int
true: ⟶bool
false: ⟶bool
suc: int ⟶int
pred: int ⟶int
insert: bag × int ⟶bag
remove: bag × int ⟶bag
element: bag × int ⟶bool

令函数集 $G=\{\varnothing$(空堆)，0，T，F. eps，+1，-1，ins(向堆中插入元素)，rem(从堆中删去元素)，?(判断某整数是否为堆中元素)$\}$对应于上述运算集。又令载体集 $A=\{A_1,A_2,A_3\}$，$A_1=\{0, 0+1, 0-1, 0+1-1, 0+1+1,\cdots\}$，$A_2=\{\varnothing, \mathrm{ins}(\varnothing,0), \mathrm{rem}(\varnothing,0), \mathrm{ins}(\mathrm{ins}(\varnothing,0+1),0),\cdots\}$，$A_3=\{T, F, ?(\varnothing,0), ?(\mathrm{rem}(\varnothing,0),0+1),\cdots\}$。则 $\overline{A}=\langle A,G\rangle$ 是对应于上述基调“整数堆”的一个 Σ 代数。

Σ 代数的层次结构　设 $\langle A,G\rangle$ 和 $\langle A',G'\rangle$ 是对应于同一基调的两个 Σ 代数。如果存在单值映射 φ，把 A 映射为 A'，G 映射为 G'，且

（1）$\varphi=\{\varphi_i \mid i \in I\} \cup \{\varphi_G\}$；

（2）对 $a_i \in A_i$，$\varphi_i(a_i) \in A_i'$，其中 A_i 和 A_i' 分别是 A 和 A' 中的载体；

（3）对 Σ 中任一运算 O_j，若 f 是 G 中与 O_j 对应的函数，则 $\varphi_G(f)=f'$ 是 G' 中与 O_j 对应的函数；

（4）由 $f(a_1,a_2,\cdots,a_k)=a_{k+1}$，其中诸 a_i 属于

A_i，可得

$$\varphi_{k+1}(f(a_1,\cdots,a_k)) = f'(\varphi_1(a_1),\cdots,\varphi_k(a_k))$$

则 φ 称为 $\langle A,G\rangle$ 到 $\langle A',G'\rangle$ 的一个 Σ 同态。Σ 同态使 Σ 代数的类子属性不变，也使它的函数结构不变。如果把 Σ 代数看作对象，把 Σ 同态看作对象间的态射，则对应于同一基调的所有 Σ 代数及其 Σ 同态构成一个范畴。

如果 Σ 代数 Σ_1 属于以某个 Σ 为基调的范畴 C，使得对 C 中的每个 Σ 代数 Σ_i 都存在一个唯一的 Σ 同态 $\varphi_i:\Sigma_1\to\Sigma_i$，则称 Σ_1 为 C 中的初始代数。如果存在 C 中的另一个 Σ 代数 Σ_2，使得对 C 中的每个 Σ_j，都存在一个唯一的 Σ 同态 $\varphi_j:\Sigma_j\to\Sigma_2$；则称 Σ_2 为 C 中的终结代数。初始代数和终结代数在同构意义下都是唯一的。在上例中，$\overline{A}$ 是整数堆的初始代数。若令 $A'_1=\{0\}$，$A'_2=[\varnothing]$，$A'_3=\{T(=F)\}$，$A'=\{A'_1,A'_2,A'_3\}$，则 $\tilde{A}=\langle A',G\rangle$ 也是对应于整数堆的一个 Σ 代数，而且是它的终结代数。

项代数和有限生成代数 任意基调 Σ 的初始代数的存在性可通过它的基项代数 $T(\Sigma)$ 的存在性来证明。$T(\Sigma)$ 定义如下：

(1) 对应于运算集 O 中的每个零目运算 O_j，$O_j:\longrightarrow s_i$ 有一个常量 O_j 属于 A_i；

(2) 若 $a_1,a_2,\cdots,a_k$ 分别属于 $A_1,A_2,\cdots,A_k$，又有运算 $O_h\in O$，$O_h:s_1\times s_2\times\cdots\times s_k\longrightarrow s_{k+1}$，则有 $O_h(a_1,a_2,\cdots,a_k)\in A_{k+1}$；

(3) 除此之外，各 A_i 无其他元素；

(4) 令 $A=\{A_i\}$，则 $T(\Sigma)=\langle A,O\rangle$；

(5) 每个 A_i 的每个元素称为一个基项。

容易证明 $T(\Sigma)$ 的存在性，以及 $T(\Sigma)$ 是初始代数。在上面的例子中，T(整数堆)和 $\overline{A}$ 同构。

一个 Σ 代数称为是有限生成的，如果它的每个元素都是用有限多个函数符号构造而成，并且不含其他符号。基项代数是有限生成代数，它的所有满 Σ 同态映象构成一个完全格。

抽象数据类型 在基项代数中允许每个载体 A_i 包含有限多个变量，经函数集 O 作用后得到含变量的项，如此生成的 Σ 代数称为项代数。一个 Σ 等式 e(又称公理)写成 $t_1=t_2$ 的形式(可以推广为更复杂的形式)，其中 t_1 和 t_2 都是项代数中的元素。对偶 $D=\langle\Sigma,E\rangle$ 称为一个抽象数据类型，其中 Σ 是一个基调，E 是一组 Σ 等式。

Σ 等式组 E 在基调 Σ 的每个 Σ 代数 $\overline{A}$ 上定义一个同余关系。即：它首先是一个等价关系。其次，对每个 $f\in G$，必能从 a_i 等价于 $b_i(1\leqslant i\leqslant n)$ 中导出 $f(a_1,\cdots,a_n)$ 等价于 $f(b_1,\cdots,b_n)$。利用 E 定义的同余关系可求出 $\overline{A}$ 的商代数 $\overline{A}/E$，它仍然是一个 Σ 代数，称为 D 的一个模型。D 的全体模型及其 Σ 同态构成一个范畴，模型 $T(\Sigma)/E$ 是此范畴中的初始代数，也称为 D 的初始模型。它表达了 D 的初始语义。D 还可以有其他的语义，如终结语义。

偏 Σ 代数 把用某种程序设计语言写的程序看成一个抽象数据类型，就可以用抽象数据类型的语义来描述程序的语义，称为代数语义。对于只有表达式计算的函数式语言，可令语言中的类型和函数分别对应于抽象数据类型中的类子和运算，并用抽象数据类型中的公理来描述类型和函数的计算规则。对于含有语句的命令式语言，可把程序的状态(各变量在每一时刻的值)作为新的类子，而语句则是作用于状态类子之上的新的运算。对于含循环和 go to 语句等动态结构的语言，可把程序的执行历史，程序执行的当前位置等动态信息作为新的类子，而把引起动态信息改变的语句结构作为新的运算。

为了表达程序中出现的不正常情况(如运算无定义、运行不终止)，在基调中引进专用函数defined。一个 Σ 等式可取下列 3 种形式：

(1) defined(t) = true；

(2) defined(t) = false；

(3) $t_1=t_2$(隐含 defined(t_1) = true；defined(t_2) = true)。

其语义是：若 defined(t) = true 在公理系统 E 中是可证的，则项 t 在该抽象数据类型 D 的任何模型中都有定义。若 defined(t) = false 是可证的，则 t 在任何模型中均无定义。若两者都不能证明，则 t 可以在有些模型中有定义，而在另一些模型中无定义。这种抽象数据类型称为偏抽象数据类型，相应的 Σ 代数称为偏 Σ 代数。例如，在抽象数据类型“整数”中，无穷大元素 ∞ 在标准模型中无定义，而在非标准模型中有定义。

Σ 同态 φ 称为是强的，若 $\varphi(t)$ 有定义当且仅当 t 有定义。φ 称为是弱的，若由 t 无定义可知 $\varphi(t)$ 一定无定义。φ 称为是共弱的，若由 t 有定义可知 $\varphi(t)$ 一定有定义。相应地可定义弱和共弱的初始模型和终结模型。已经证明，任何一个偏抽象数据类型 D 都有一个共弱初始模型和弱终结模型。如果 D 是描述程序语义的，那么此弱终结模型与该程序的最小不动点语义等价。

参考文献

陆汝钤. 计算系统的形式语义. 北京:清华大学出版社,2017 (陆汝钤)

lingxing wenfa

0 型文法(type 0 grammar) 参见短语结构文法。

yixing wenfa

1 型文法(type 1 grammar) 参见上下文有关文法。

erxing wenfa

2 型文法(type 2 grammar) 参见上下文无关文法。

sanxing wenfa

3 型文法(type 3 grammar) 参见正则文法。

3GPP changqi yanjin

3GPP 长期演进(long term evolution, LTE)

GSM 超越 3G 和 HSDPA 阶段迈向 4G 的进阶版本,它由第三代合作伙伴计划(The 3rd Generation Partnership Project, 3GPP)标准组织制定。LTE 也被俗称为 3.9G。2010 年 12 月 6 日国际电信联盟把 LTE 正式称为 4G。

LTE 系统的关键技术

(1) 物理层多址传输技术 下行采用正交频分多址接入(orthogonal frequency division multiple access, OFDMA),上行采用单载波频分多址(single carrier-frequency division multiple access, SC-FDMA)。正交频分复用(orthogonal frequency division multiplexing, OFDM)技术是 LTE 系统的技术基础与主要特点。其中,载波间隔又是 OFDM 系统的最基本参数,经理论分析和仿真比较最终确定为 15 kHz。上、下行的最小资源块为 180 kHz,即 12 个子载波宽度。数据到资源块的映射方式可采用集中方式或分布方式。

(2) 天线技术 采用多输入多输出(multiple-input multiple-output, MIMO)技术。LTE 已确定 MIMO 天线个数的基本配置是下行 2×2,上行 1×2,但也考虑 4×4 的天线配置。LTE MIMO 下行方案可分为两大类,即发射分集和空间复用。目前,考虑采用的发射分集方案包括块状编码传送分集、时间(频率)转换发射分集、包括循环延迟分集在内的延迟分集以及基于预编码向量选择的预编码技术。

(3) 编码和调制 采用高阶调制和 Turbo 码。LTE 下行主要采用 QPSK、16QAM、64QAM 三种调制方式。上行主要采用位移 BIT/SK、QPSK、8PSK 和 16QAM。在信道编码方面,LTE 主要考虑 Turbo 码,但如果能获得明显的增益,也将考虑其他编码方式,如 LDPC 码。为了实现更高的处理增益,还可以考虑以重复编码作为前向纠错码(FEC)的补充。

LTE 技术特点

(1) 提高了通信效率和频谱效率 系统最大宽带为 20 MHz,在这样的宽带下,下行峰值速率为 100 Mb/s、上行峰值速率为 50 Mb/s。

(2) 分组交换与服务质量(QoS)保证 系统在整体构架上架基于分组交换,同时通过系统设计和严格的 QoS 机制,保证实时业务的服务质量。

(3) 支持各种系统宽带 除了 20 MHz 最大宽带以外,还能够支持 1.4 MHz、3 MHz、10 MHz 和 15 MHz 等系统宽带,以及“成对”与“非成对”频段的部署,以保证将来在系统部署上的灵活性。

(4) 明确提出系统在支持高移动速率的基础上,需要考虑为低移动速率用户提供优化条件,同时改善小区边缘用户的吞吐量等具体的系统需求。

参考文献

1. 王映民, 孙韶辉,等. TD-LTE 技术原理与系统设计. 北京:人民邮电出版社, 2010

2. 曾召华. LTE 基础原理与关键技术. 西安:西安电子科技大学出版社, 2010 (张平)

条目汉语音序索引

说　明

1. 本索引供读者按条题名的汉语拼音寻查条目。为了保证全书条目的完整性，外文条题名的条目也收入本索引内。
2. 本索引按条题名的汉语拼音字母顺序排列。第一字同音时，按四声顺序排列；同音同调时按笔画多少和笔顺排列；如完全相同，则按第二字，余类推。
3. 外文条题名排在最后，依次按英文字母、希腊字母和阿拉伯数字顺序排列。

A

B

C

D

E

F

G

H

J

K

S

Z

INDEX OF ARTICLES

(条目外文索引)

说　明

本索引按条目的外文条题名的英文字母顺序排列。非英文字母的条题名则按希腊字母及阿拉伯数字的顺序排在后面。

A

B

C

D

E

F

I

N

O

P

Q

R

S

T

U

V

内 容 索 引

说　明

1. 本索引是全书条目和释文中的主题词索引。索引主题按汉语拼音字母的顺序排列。第一字同音时,按其音调四声顺序排列;同音同调时依次按笔画多少和笔顺排列;如完全相同,则按第二字,余类推。非汉字开头的索引主题排在汉字主题后。依次为英文字母、希腊字母和阿拉伯数字开头的索引主题,它们分别按其字母顺序和数的顺序排列。

2. 设有条目的主题用黑体字印出,未设条目的主题用仿宋体字印出。

A

B

D

E

F

G

H

J

K

L

M

N

P

Q

R

S

T

X

Y

Z

A

B

C

D

E

F

G

H

I

J

附录 I　缩略语

缩略语	英文全名	中文译名
AAD	analog alignment diskette	模拟校准盘
AAL	ATM adaptive layer	ATM 适配层
ABI	application binary interface	应用二进制接口
ABM	asynchronous balanced mode	异步平衡模式
ABR	alternate bit rate	自适应式比特率(服务)
ABS	Agfa balanced screening	Agfa[公司]均衡挂网(技术)
AbS	analysis-by-synthesis	合成-分析法
AC	access control	访问控制
ACID	atomicity, consistency, isolation and durability	不可再分割性,一致性,隔离性与耐用性
ACL	agent communication language	主体通信语言
ACL	Association for Computational Linguistics	计算语言学学会(美国)
ACM	Association for Computing Machinery	计算机协会(美国)
ACSE	association control service element	联系控制服务要素
ACSL	advanced continuous simulator language	高级连续仿真语言
ADC	analog-to-digital converter	模数转换器
ADL	architecture description language	体系结构描述语言
ADM	add/drop multiplxer	多路复用与多路分解器
ADPCM	adaptive differential pulse code modulation	自适应差分脉[冲编]码调制
ADSL	asymmetric digital subcriber line	非对称数字用户线
AES	application environment specification	应用环境规范
AET	active edge table	有效边表
AF	assured forwarding	保证转发
AFS	Andrew File System	(卡内基-梅隆大学的)分布式文件系统
AFS	automatic file system	自动文件系统
AGC	automatic gain control	自动增益控制
AH	authentication leader	鉴别报头
AHPL	a hardware programming language	硬件程序设计语言,AHPL 语言
AI	artifitial intelligence	人工智能

缩略语	英文全名	中文译名
AIT	advanced intelligent tape	高级智能磁带(技术)
AIX	advanced interactive executive	高级交互执行程序
ALGOL	algorithmic language	算法语言
ALICE	automated location of isolation and continuity errors	孤立与连续性错误的自动定位
ALOA	asset library open architecture framework	开放体系结构的构件库框架
ALPAC	Automatic Language Processing Advisory Committee	语言自动处理咨询委员会
ALU	arithmetic and logic unit	算术逻辑部件,运算器
AM	address mark	地址标志
AMA	abstract muscle action	抽象肌肉动作
AMBIT	Acronym May Be Ignored Totally	AMBIT(语言)
AMD	Advanced Micro Devices Inc.	先进微型器件公司,AMD 公司
AME	super metal evaporation	高级金属蒸镀带(技术)
AMF	address mark found	找到地址标志
AMP	advanced metal evaporation	高级金属粉末
ANN	artificial neural network	人工神经网络
ANS	Advanced Network Services, Inc.	先进网络服务公司
ANSA	Advanced Network System Architecture	先进网络体系结构(日本东芝公司)
ANSI	American National Standards Institute	美国国家标准学会
AOP	agent-oriented programming	面向主体的程序设计
AP	access point	访问点
AP	array processor	阵列处理机
API	application programming interface	应用程序设计接口
APL	A Programming Language	APL[程序设计]语言
APON	ATM passive optical network	异步传输模式无源光纤网
APPANET	Advanced Research Project Agency Network	(美国)APPA 网
APT	Automatically Programmed Tools	自动程控机床(语言),APT(语言)
AR	augmented reality	增强现实
ARGUS	automatic routine generating and updating system	自动例程生成和更新系统
ARM	asynchronous response mode	异步响应模式
ARP	address resolution protocol	地址解析协议
ARPA	Advanced Research Projects Agency	(美国)高级研究计划署
ARQ	automatic repeat request	停-等自动重复请求
ART	adaptive resonance theory	自适应谐振理论
AS	activity scanning	活动扫描(法)
AS	autonomous system	自治系统
ASA	Advanced Software Automation	高级软件自动化(公司)

缩略语	英文全名	中文译名
ASC	American Standards Committee	美国标准委员会
ASCI	Accelerated Strategic Computing Initiative	战略的加速计算倡导者联合会
ASCII	American standard code for information interchange	美国信息交换标准[代]码,ASCII 码
ASIC	application specific integrated circuit	专用集成电路
ASIC	application specified integrated circuit	按应用需求定制的集成电路
ASK	amplitude shift keying	幅移键控
ASK	a simple knowledgeable system	一种简单知识系统
ASL	American sign language	美国符号语言
ASLIC	application specific logic integrated circuit	专用逻辑集成电路
ASN. 1	abstract syntax notation. one	抽象句法表示法 1
ASP	application service provider	应用服务提供商
ASPOL	a simulation process-oriented language	一种面向模拟过程语言
ASR	automated send/receive	自动发送接收
ATA	advanced technology attachment	高级技术配件
ATAPI	advanced technology attachment packet interface	高技术配件分组接口
ATC	address translation cache	地址转换高速缓存
ATC	air traffic control	空中交通指挥
ATE	automatic test equipment	自动测试设备
ATM	asynchronous transfer mode	异步传送模式
ATM	automatic teller machine	自动出纳机
ATMLAN	asynchronous transfer mode local area network	异步传送模式局域网
ATN	augmented transition network	扩充转移网络,增强型转移网络
ATP	automatic theorem proving	自动定理证明
AU	access unit	访问单元
AUC	authentication center	鉴权中心
AUI	attachment unit interface	连接部件接口
AURA	automated reasoning assistant	自动推理助理(系统),AURA(系统)
AVS	application visualization system	应用可视化系统
BAF	branch address field	转移地址字段
BAS	building automatic system	建筑物自动化系统
BASIC	Beginner's All-purpose Symbolic Instruction Code	初学者通用符号指令码(语言),BASIC(语言)
BBN	Bolt, Beranck and Newman, Inc.	博尔特·贝拉尼克和纽曼公司
BBS	bulletin board system	公告板系统
B2C	business to consumer	商家到客户
BCD	binary-coded decimal	二-十进制

缩略语	英文全名	中文译名
BCF	branch control field	转移控制字段
BCH	Bose-Chaudhuri-Hocqueghem (code)	博斯-乔赫里-霍克文黑姆(码),BCH(码)
BCPL	basic combined programming language	基本的组合式程序设计语言
BDI	belief-desire-intention	信念-期望-意图
BDI	business data interchange	商务数据交换
BECN	backward explicit congestion notification	后向显示拥塞控制标志
Bellcore	Bell Communication Research	贝尔通信研究所
BFL	buffered field effect transistor logic	缓冲场效应晶体管逻辑电路
BGA	ball grid array package	球栅阵列封装
BG	black generation	黑色生成函数
BGP	Border Gatway Protocol	边界网关协议
BHM	basic hypermedia model	基本超媒体模型
BICFET	bipolar inversion channel field effect transistor	双极反型沟道场效应晶体管
BIDM	basic interoperability data model	基本互操作数据模型
BiFET	bipolar field effect transistor	双极型场效应晶体管
BiMOS	bipolar MOS	双极型金属-氧化物-半导体
BIOS	basic input/output system	基本输入输出系统(管理程序)
B-ISDN	broadband integrated services digital network	宽带综合业务数字网,宽带 ISDN
BIST	built-in self-test	内装自测试,内建自测试
BLAS	basic linear algebra subprogram	基本线性代数子程序
BLISS	basic language for the implementation of system software	实现系统软件的基本语言
BLOB	binary large object	二元大客体
BMP	basic multilingual plane	基本多文种平面
BNA	Burroughs Network Architecture	宝来网络体系结构(美国 Burroughs 公司)
BNF	Backus-Naur form alism	巴克斯-诺尔形式体系
BNF	Backus normal form	巴克斯范式
BP	back propagation	BP 算法,反传
BPF	band pass filter	带通滤波器
BPR	business process reengineering	企业流程重组
BPSK	binary phase shift keying	二进制相移键控
B-rep	boundary representation	边界表示法
BRI	basic-rate interface	基本速率接口
BS	base stations	基站

缩略语	英文全名	中文译名
BSC	base station control	基站控制器
BSD	Berkeley software distribution	(美国加州)伯克利软件发行版本
BSP	board support package	板支持组件
BSP	Burroughs Scientific Processor	宝来(公司)科学处理机
BSS	base station subsystem	基站子系统
BTB	branch target buffer	转移目标缓冲
BTC	base transceiver stations	基站收发信机
CA	certificate authority	证书权威(机构)
CACI	Consolidated Analysis Centers, Inc.	联合分析中心
CACM	Communications of the Association for Computing Machinery	(美国)计算机协会通信
CADAM	computer-graphics-augmented design and manufacturing	计算机图形扩充设计与制造
CAD/CAM	computer-aided design/computer-aided manufacturing	计算机辅助设计和制造
CAD	computer-aided design	计算机辅助设计
CADDS	computer-aided design and drafting system	计算机辅助设计与制图系统
CAE	computer-aided engineering	计算机辅助工程
CAFS	content addressable file storage	按内容寻址文件存储器
CAGD	computer-aided geometric design	计算机辅助几何设计
CAI	computer-assisted instruction	计算机辅助教学
CALS	commercial at light speed	光速商务
CAM	cellular automata machine	细胞自动机
CAM	computer-aided manufacturing	计算机辅助制造
CAM	computer-aided mapping	计算机辅助地图绘制
CAMI	automated process planning system	自动工艺规划系统
CAPP	computer-aided process planning	计算机辅助工艺规划(系统)
CAQ	computer-aided quality control	计算机辅助质量控制
CASE	common application service element	公共应用服务要素
CASE	computer-aided software engineering	计算机辅助软件工程
CAT	computer-aided testing	计算机辅助测试
CAT	computer-aided test	计算机辅助检测
CATIA	computer-graphics-aided three-dimensional interactive application	计算机图形辅助三维交互应用
CATV	cable television	电缆电视
CAV	constant angular velocity	恒角速度
CAVE	computer automatic virtual environment	洞穴式显示装置(计算机自动虚拟环境)

缩略语	英文全名	中文译名
CBE	computer-based education	计算机辅助教育
CBL	computer-based learning	计算机辅助学习
CBR	case-based reasoning	基于案例的推理
CBR	constant bit rate	恒定比特率(服务)
CBSD	component-based software development method	基于构件的软件开发方法
CBSE	component-based software engineering	基于构件的软件工程
CBT	computer-based teaching	计算机辅助教学
CBX	computerized branch exchange	计算机化小交换机
CCA	common cryptography architecture	公共密码体系[结构]
CCA	Computer Corporation of America	美国计算机公司
CC	common criteria of IT security evaluation	通用安全评估准则
CCD	charge coupled device	电荷耦合器件
CC-DOS	Chinese Character Disk Operating System	汉字磁盘操作系统
CCFL	charge-coupled FET logic	电荷耦合场效应晶体管逻辑电路
CCIR	Consultative Committee on International Radio	国际无线电咨询委员会
CCITT	Consultative Committee on International Telegraph and Telephone	国际电话电报咨询委员会
CC-NUMA	cache coherency non-uniform memory access	高速缓冲存储器一致性非均匀存储器访问
CCS	calculus of communicating system	通信系统演算(语言),CCS(语言)
CCSS	common channel signalling system	公共信道信令系统
CCTA	Central Computer and Telecommunications Agency	中央计算机和电信总局(英国)
CCU	communications control unit	通信控制器
CDDI	copper distributed data interface	铜线分布式接口
CDE	common desktop environment	公共桌面环境
CD-G	CD-graph	CD-G 视盘
CD-I	CD-interactive	交互式激光视盘
CDI	compact disk interactive	交互式光盘
CDL	component definition language	构件定义语言
CDL	computer design language	计算机设计语言
CDMA	code division multiple access	码分多址接入
CD-R	compact disc-recordable	数字可写光盘
CD-ROM	compact disc-read only memory	只读光盘,CD-ROM 盘
CD-RW	compact disc-rewritable	数字可重写光盘
CD-V	CD-video	CD-V 视盘
CEC	Commision of the European Communities	欧洲共同体委员会

缩略语	英文全名	中文译名
CE	concurrent engineering	并行工程
CE	customer engineer	现场工程师
CEDAR	computer-aided environmental design, analysis and realization	计算机辅助环境设计,分析和实现
CEGA	Chinese enhanced graphics adapter	汉字 EGA 图形适配器
CEL	current events list	当前事件表
CELP	code excited linear prediction coding	码激励线性预测编码
CELP	code excited linear predictive	码激励线性预测法
CEPT	Conference of European Postal and Telecommunications Administrations	欧洲邮政电信管理会议
CF	certainty factor	确信度
CF	compact flash	袖珍闪存
CFG	context-free grammar	上下文无关文法,2 型文法
CFL	context-free language	上下文无关语言
CFS	computer file system	计算机文件系统
CGA	colour graphics adapter	彩色图形适配器
CG	computer graphics	计算机图形学
CGI	common gateway interface	公共网关接口
CGI	Computer Graphics Interface Standard	CGI 图形接口标准
CGM	Computer Graphics Metafile Standard	图形元文件标准
CGRM	Computer Graphics Reference Model	计算机图形基准模型
CIC	Computer Information Center Ltd.	计算机信息中心有限公司,CIC 公司
C3I	Command, Control, Communication and Intelligence	指挥、控制、通信及情报系统
C4I	Command, Control, Communication, Computer and Intelligence	指挥、控制、通信、计算机及情报
CICS	client information control system	客户信息控制系统
CIF	common intermediate formal	公用中分辨率图像格式
CIFS	common internet file system	通用互连文件系统
CIM	computer integrated manufacturing	计算机集成制造
CIMS	computer-integrated manufacturing system	计算机集成制造系统
CIP	Chinese information processing	中文信息处理
CIP	computer-aided, intuition-guided programming	计算机辅助–直觉指导的程序设计
CIRC	cross interleave Reed-Solomon code	交叉交错里德–索罗门码
CISC	complex instruction set computer	复杂指令集计算机
C4ISR	Command, Control, Communication, Computer, Intelligence, Surveillance and Reconnaissance	指挥、控制、通信、计算机、情报、监视及侦察
CITIS	contractor integrated technical information service	主承包商技术信息集成服务

缩略语	英文全名	中文译名
CIX	commercial Internet exchange	商用 Internet 交换
CJK	China-Japan-Korea	中日韩
CLA	carry-lookahead adder	先行进位加法器
CLFL	complement level field effect transistor	互补电平场效应晶体管
CLIPS	C language integrated production system	C 语言综合生产系统
CLNP	connectionless-mode network protocol	无连接方式网络协议
CLNS	connectionless-mode network service	无连接方式网络服务
CLOS	CLOSE statement processor	CLOSE 语句处理程序
CLP	cell loss priority	信元丢失优先级
CLP	constraint logic programming	约束逻辑程序设计(语言)
CLS	conceptual learning system	概念学习系统
CLV	constant linear velocity	恒线速度
CMAC	cerebellar model articulation controller	小脑网络模型
CMAR	control memory address register	控存地址寄存器
CM	control memory	控制存储器,控存
CMI	computer-managed instruction	计算机管理教学
CMIP	Common Management Information Protocol	公共管理信息协议
CMIS	common management information service	公共管理信息服务
CMISE	common management information service element	公共管理信息服务要素
CML	Chemical Markup Language	化学置标语言
CMM	capability maturity model	能力成熟度模型
CMMI	Capability Maturity Model Integration	能力成熟度模型集成(项目)
CMOS	complementary metal-oxide-semiconductor	互补金属氧化物半导体
CMOT	common management information service and protocol on TCP/IP	公共管理信息服务和基于 TCP/IP 的协议
CMP	chemical mechanical polish	化学机械抛光
CMY	cyan-magenta-yellow	青-品红-黄(模型)
CMYK	cyan-magenta-yellow-black	青-品红-黄-黑(模型)
CNC	computer numerical control	计算机数值控制,计算机数控
COB	chip on board	板上芯片(封装)
COBOL	common business-oriented language	面向商业的通用语言
COBUILD	COLLINS Birmingham University International Language Database	COLLINS 伯明翰大学国际语言数据库
CO-CAD	cooperative computer aid design	协同计算机辅助设计
CO	central office	中心局
CODASYL	Conference on Data System Languages	数据系统语言研究会

缩略语	英文全名	中文译名
COMA	cache-only memory architecture	惟高速缓存存储体系结构
COM	Component Object Model	(微软)构件对象模型
COMIT	compiler of Massachusetts Institute of Technology	麻省理工学院编译程序(语言)
CORBA	Common Object Request Broker Architecture	公共对象请求代理体系结构
COS	cooperation for open systems	开放系统协作
COSE	Common Open Software Environment	公共开放软件环境(组织)
COSE	common open system environment	通用开放系统
COTS	commercial-off-the-shelf	货架商品式(构件)
COTS	commodity off the shelf	流行商品
CPE	customer premises equipment	用户宅院设备
CPI	cycles per instruction	执行每条指令的平均周期数
CPM	critical path method	关键路径方法
CPO	complete partial order	完全偏序
CPU	central processing unit	中央处理机,中央处理器
CRC	cyclic redundancy check	循环冗余检验
CRCW	concurrent read concurrent write	并发读写
CREW	concurrent read exclusive write	并发读互斥写
CRL	certificate revoke list	证书撤销列表
CRT	cathode ray tube	阴极射线管
CRTC	cathode ray tube controller	阴极射线管控制器,CRT 控制器
CS	convergence sublayer	会聚子层
CSA	carry save adder	保留进位加法器
CSCW	computer-supported cooperative work	计算机支持的协同工作,群体计算
CSG	constructive solid geometry	构造的实体几何
CSG	context sensitive grammar	上下文有关文法
CSL	context sensitive language	上下文有关语言
CSLI	Center for the Study of Language and Information	语言和信息研究中心(美国 Stanford 大学)
CSMA/CD	carrier sense multiple access with collision detection	带碰撞检测的载波侦听多址访问
CSM	command service module	命令服务模块
CSP	chip scale package	芯片尺寸封装
CSP	Communication Sequential Processing	通信顺序处理(语言),CSP(语言)
CSP	communicating sequential process	通信顺序进程
CSP	cross system product	交互系统产品
CSS	cascading style sheets	级联样式表

缩略语	英文全名	中文译名
CSS	chunk self-scheduling	块[自]调度
CSS	computer system simutator	计算机系统模拟器
CSS	contact start stop	接触启停(技术)
CSSL	continuous system simulation language	连续系统仿真语言
CSU	channel service unit	信道服务部件
CT	computer tomography	计算机断层摄影
CTP	composite theoretical performance	综合理论性能
C-T-R	Calculating,Tabulating and Recording Co.	计算、制表与录制公司,C-T-R 公司
CVC	call virtual circuit	呼叫虚电路
CVD	chemical vapor deposition	化学气相淀积
CVS	concurrent versions system	并行(软件)版本控制系统
CWA	closed-world assumption	封闭世界假设
DA	design automation	设计自动化
DA	destination address	目的地址
DAC	digital-to-analog converter	数模转换器
DAISY	dynamically architected instruction set from Yorktown	Yorktown 动态构造指令集
DAM	direct access memory	直接存取存储器
DARPA	Defense Advanced Research Projects Agency	国防高级研究计划署(美国防部)
DAS	direct-attached storage	直接连接存储
DAT	digital audio tape	数字音频磁带
DB	database	数据库
DBA	database administrator	数据库管理员
DBCLI	Database Call Level Interface	数据库调用级接口(草案)
DBCS	double byte character sets	双字节字符集
DBMS	database management system	数据库管理系统
DBS	direct broadcast systems	卫星直播系统
DBTG	Data Base Task Group	数据库任务组
DC	device coordinate system	设备坐标系
DCE	data circuit equipment	数据电路设备
DCE	distributed computing environment	分布式计算环境
DCFL	direct coupled field effect transistor logic	直接耦合场效应晶体管逻辑电路
DCG	definite clause grammar	定子句语法
DCH	dependence convex hull	依赖凸边域(方法)
DCNA	data communication network architecture	数据通信网络体系结构
DCOM	distributed component object model	分布式构件对象模型
DCS	distributed control system	集散控制系统

缩略语	英文全名	中文译名
DCSL	deterministic context sensitive language	确定型上下文有关语言
DCT	discrete cosine transform	离散余弦变换
DD	data dictionary	数据词典
DDA	digital differential analyzer	数字微分分析仪
DDBMS	distributed data base management system	分布式数据库管理系统
DDC	direct digital control	直接数字控制
DDD	digital diagnostic diskette	数字诊断盘
DDE	dynamic data exchange	动态数据交换
DDI	data display indicator	数据显示指示器
DDI	direct digital interface	直接数字接口
DDL	data definition language	数据定义语言
DDL	digital system design language	数字系统设计语言
DDL	domain description language	领域描述语言
DDMS	distributed database management system	分布式数据库管理系统
DDN	digital data network	数字数据网
DDR	double data rate	双倍数据速率
DDR SDRAM	double data rate SDRAM	双倍速率同步动态随机存储器
DE	discard eligibility	适合丢弃
DEC	Digital Equipment Corp.	数字设备公司,DEC 公司(美国)
DEDALUS	deductive algorithm Ur-synthesizer	演绎算法 Ur 综合器
DEDS	discrete event dynamic system	离散事件动态系统
DENDRAL	dendritic algorithm	树枝状算法
DES	data encryption standard	数据加密标准
DES	distributed expert system	分布式专家系统
DF	don't fragment	禁止分段
DFA	deterministic finite automaton	确定型有限自动机
DFD	data flow diagram	数据流程图
DFG	data flow graph	数据流图
DFSA	deterministic finite state automaton	确定型有限[状态]自动机
DFT	design for testability	可测性设计
DFT	discrete Fourier transform	离散傅里叶变换
DI	data input	数据输入
DIPS	Dhrystone instructions per second	条 Dhrystone 指令每秒
DIS	distributed interactive simulation	分布交互式仿真
DIS	draft international standard	国际标准草案
DLAT	directory lookaside table	目录检测表

缩略语	英文全名	中文译名
DLBA	deterministic linear bounded automaton	确定型线性有界自动机
DLCI	data link connection identifier	数据链路连接标识符
DLT	digital linear tape	数字线性磁带(技术)
DM	data mining	数据挖掘
DM	delay modulation	延迟调制(码)
DMA	direct memory access	直接存储器存取
DME	distributed management environment	分布式管理环境
DML	data manipulation language	数据操纵语言
DMRP	distributed manufacturing resource planning	分布式制造资源规划
DMS	data management system	数据管理系统
DNA	deoxyribose nucleic acid	脱氧核糖核酸
DNA	Digital Network Architecture	数字网络体系结构(美国 DEC 公司)
DNC	direct numerical control	直接数值控制,直接数控
DO	data output	数据输出
DOC	diskonchip	片上固态盘
DocBook	document book	电子书
DOCF	data operation control field	[数据]操作控制字段
DOCSIS	data over cable service interface specification	通过电缆服务接口传输数据规范
DOD	Department of Defense	国防部(美国)
DOES	digital optoelectronic switch	数字光电开关
DOK	diskonkey	加密固态盘
DOM	diskonmodule	模板上固态盘
DOM	document object model	文档对象模型
DOR	dimension oriented routing	按维寻径
DOS	Disk Operating System	磁盘操作系统
DPCM	differential pulse code modulation	差分脉码调制
DPDA	deterministic pushdown automaton	确定型下推自动机
DPSK	diffirential phase shift keying	差分相移键控
DQDB	distributed queue dual bus	分布式队列双总线
DRAM	dynamic random access memory	动态随机存储器
DRM	distributed resource management	分布的资源管理
DS	network differential serices	网络区分服务
DSC	decision support center	决策支持中心
DS3	digital signal level 3	第 3 级数字信号(数据传输速率)
DSI	data stream interface	数据流接口(速率)
DSL	data subscriber line	数字用户专用线

缩略语	英文全名	中文译名
DSM	distributed shared memory	分布式共享存储器,分布式共享主存
DSP	digital signal processing	数字信号处理
DSP	digital signal processor	数字信号处理器
DSS	decision support system	决策支持系统
DSSA	domain specific software architecture	领域特定的软件体系结构
DSSD	data structured system development	数据结构化系统开发
DSSS	direct-sequence spread-spectrum	直接序列扩频
DSV	digital sum variation	“数字和”变化量
DTD	document type definition	文档类型定义
DTE	data terminal equipment	数据终端设备
DTL	diode-transistor logic	二极管-晶体管逻辑电路
DTMF	dual tone multiple frequency	双音多频
DTP	desktop publishing system	桌面出版系统
DUT	device under test	被测器件
DVD	digital versatile disc	数字多用途光盘
DVI	digital video interactive	交互式数字视频,交互式数字视频(系统)
DXF	Drawing Graphics Exchange format	绘图交换文件格式
EAN	European article number	欧洲商品(条码),EAN(条码)
EARC	Extraordinary Administrative Radio Conference	无线电特别行政会议
EAROM	electrically alterable PROM	电可修改只读存储器
EB	electronic beam	电子束
EBCDIC	Extended Binary Coded Decimal Interchange Code	扩充的二-十进制交换码,EBCDIC 码
EBL	explanation-based learning	基于解释学习
EC	electronic commerce	电子商务
ECC	error correcting code	纠错码
ECIRC	European Computer Industry Research Center	欧洲计算机工业研究中心(德国)
ECL	emitter coupled logic	射极耦合逻辑[电路]
ECMA	European Computer Manufacturers Association	欧洲计算机制造商协会
ECML	electronic-commerce model language	电子商务模型语言
ECSS	extendible computer system simulator	可扩充的计算机系统模拟器
EDA	electronic design automation	电子设计自动化
EDB	extensional database	外延数据库
EDC/ECC	error detection/error correction code	检纠错码
EDI	electronic data interchange	电子数据交换
EDI Ⅱ	enterprise data integration	企业数据集成

缩略语	英文全名	中文译名
EDIMS	EDI messaging system	EDI 消息处理系统
EDIN	EDI notification	EDI 通知
EDI-UA	EDI user agent	EDI 用户代理
EDO	extended data output	数据扩展输出
EDSAC	electronic discrete sequential automatic computer	电子离散时序自动计算机,EDSAC 计算机
EDVAC	electronic discrete variable automatic computer	电子离散变量自动计算机,EDVAC 计算机
EEPROM	electrically erasable programmable read only memory	电可擦[可]编程只读存储器
EER	equal error rate	等错误率点
EF	expedited forwarding	加速转发
EFM	eight-fourteen modulation	8/14 调制(码)
EGA	enhanced graphics adapter	增强型彩色图形适配器
EGP	exterior gateway protocol	外部网关协议
EIA	Electronic Industries Association	(美国)电子工业协会
EIDE	enhanced IDE	增强型集成驱动电子线路接口
EIR	equipment identity register	设备识别寄存器
EISA	extended industry standard architecture	扩展工业标准体系结构
EJB	Enterprise Javabeans	企业 Java 组件
EL1	extensible language 1	可扩充语言 1
EMC	electromagnetic compatibility	电磁兼容[性]
EMI	electromagnetic interference	电磁干扰
EMSP	enhanced modular signal processor	增强型模块化信号处理器
ENIAC	electronic numerical integrator and calculator	电子数字积分器和计算器,ENIAC 计算机
ENRZ1	enhanced non-return-to-zero change on one	增强不归零 1 制
ENRZ	enhanced non-return-to-zero	增强不归零制
EPIC	explicitly parallel instruction computing	显式并行指令计算
EPL	Experimental Programming Language	试验性程序设计语言
EPROM	erasable programmable read only memory	可擦[可]编程只读存储器
EREW	exclusive read exclusive write	互斥读写 X
ERP	enterprise resource planning	企业资源规划
ESCON	Enterprise System Connection Architecture	(IBM 公司)企业系统连接体系结构
ESDI	enhanced small device interface	增强型小设备接口,ESDI 接口
ES	end system	端系统
ES	event scheduling	事件调度(法)
ESP	encapsulation security payload	封装安全负载

缩略语	英文全名	中文译名
ESPRDIT	European Strategic Programme for Research and Development in Information Technology	欧洲信息技术战略研究和发展计划
ET	edge table	边表
ETS	expertise transfer system	专业知识转换系统
ETSI	European Telecommunication Standards Institute	欧洲电信标准协会
EUC	extended UNIX code	扩展的 UNIX 码
EXCA	exchangeable card architecture	可互换板卡体系结构
FACS	facial action coding system	面部动作编码系统
FA	factory automation	工厂自动化
FAR	false acceptance rate	错误接受率
FAT	file allocation table	文件分配表
FC	fiber channel	光纤通道
FC	flip-chip	倒装焊
FC	frame control	帧控制
FCS	fieldbus control system	现场总线控制系统
FCS	frame check sequence	帧检验序列
FD	frequency doubling	倍频(制)
FDD	floppy disk drive	软磁盘驱动器
FDDI	fiber distributed data interface	光纤分布式数据接口
FDMA	frequence division multiple access	频分多址接入
FDM	frequency division multiplexing	频分多路复用
FECN	forward explicit congestion notification	前向显式拥塞通知
FEL	future events list	将来事件表
FEP	front-end processor	前端处理器
FET	field effect transistor	场效应晶体管
FFD	fast flash disk	快闪固态盘
FFM	form flow model	表格流模型
FFS	fast file system	快速文件系统
FFT	fast Fourier tansform	快速傅里叶变换
FGSPEC	functional graphical specification	函数图解规约(语言)
FHSS	frequency-hopping spread-spectrum	跳频扩频
FIFO	first in, first out	先进先出
FILO	first in, last out	先进后出
FLAIR	functional language articulated interactive resource	函数式语言接合的交互资源
FM	frequency modulation	调频(制)
FM-AM	file management-access method	文件管理存取方法

缩略语	英文全名	中文译名
FMP	file management protocol	文件管理协议
FMP	flow model processor	流程模型处理机
FMR	false match rate	错误匹配率
FMS	file management system	文件管理系统
FMS	flexible manufacturing system	柔性制造系统
FNMR	false non match rate	错误非匹配率
FORMAC	Formula Manipulation Compiler	公式处理编译程序(语言)
FORTRAN	Formula Translation	公式翻译(语言)
FP	functional programming	函数式程序设计
FPGA	field programmable gate array	现场可编程门阵列
FPLA	field-programmable logic array	现场可编程逻辑阵列
FPM	fast page mode	快速页面模式
FPM	FTAM protocol machine	FTAM 协议机,文件传送、存取和管理协议机
FPS	fast packet switching	快速分组交换
FPS	floating-point system	浮点系统
FR	frame relay	帧中继
FRAM	ferroelectron RAM	铁电随机存取存储器
FS	frame state	帧状态
FSA	finite state automaton	有限[状态]自动机
FSK	frequency shift keying	频移键控
FSM	finite state machine	有限状态自动机
FTAM	file transfer, access and management	文件传送、存取和管理
FT-AM	file transfer-access method	文件传送存取方法
FTP	File Transfer Protocol	文件传送协议
FTTC	fiber-to-the-curb	光纤到路边
FUG	functional unification grammar	功能合一语法
GA	gate array	门阵列
GaAsIC	GaAs integrated circuit	砷化镓集成电路
GaAsMOS	GaAs MOS	砷化镓 MOS
GAL	generic array logic	通用阵列逻辑[电路]
GASP	general activity simulation program	通用活动仿真程序
GCD	greatest common divisor	最大公约数
GCP	ground control point	地面控制点
GCR	group coded recording	成组编码记录
GDI	graphics device interface	图形设备接口

缩略语	英文全名	中文译名
GDSS	group decision support system	群体决策支持系统
GEO	geostationary earth orbit	对地静止轨道
GFC	generic flow control	一般流量控制
GGCA	Gemetric Graphics Content Architecture	几何图形内容体系结构
GGF	Global Grid Forum	全球网格论坛
GHC	guarded Horn clause	保护性霍恩子句
GII	global information infrastructure	全球信息基础设施
GIIT	graphical input interactive technique	图形输入交互技术
GIS	geographic information system	地理信息系统
GKS-3D	graphical kernel system for three dimensions	三维图形核心系统
GKS	graphical kernel system	图形核心系统
GKSM	GKS metafile	GKS 元文件(接口);GKS 图形标准
4GL	fourth generation language	第四代语言
GLSI	gigantic large scale integration	巨大规模集成[电路]
GMD	Gesellschaft fur Mathematik und Datenverbetung	德国政府科学研究中心
GMM	glued multi-resolution model	黏合多分辨率模型
GOOD	general object-oriented software development	通用的面向对象软件开发
GOSIP	government open systems interconnection profile	政府开放系统互连功能轮廓
GP	graphic processor	图形处理器
GPIB	general purpose interface bus	通用接口总线
GPIO	general purpose input/output	通用输入输出(接口)
GPS	general problem solving system	通用问题求解系统
GPS	general problem solving	一般问题求解
GPS	global position system	全球定位系统
GPSG	generalized phrase structure grammar	广义短语结构语法
GPSS	general purpose systems simulation	通用系统模拟(语言),GPSS(语言)
GPSS	general purpose systems simulator	通用系统仿真器
GRASS	geographical resources analysis support system	地理资源分析支持系统
GRIP	graphics interactive programming	图形交互程序设计
GSM	Global System for Mobile Communication	全球移动通信系统
GSS	guided self-scheduling	引导[自]调度
GT	group technology	群[成组]技术
GUI	graphical user interface	图形用户界面
GUIDS	graphical user interface description system	图形用户界面描述系统
GUSTO	Globus Uniquitous Supercomputing Testbed Organization	Globus 随遇超级计算试验床组织
GWUIMS	George Washington User Interface Management System	乔治·华盛顿用户界面管理系统

缩略语	英文全名	中文译名
HAL	hardware abstraction layer	硬件抽象层
HAMT	human-aided machine translation	人助机译
HBT	heterojunction bipolar transistor	异质结双极[型]晶体管
HCI	human computer interaction	人机交互(接口)
HCSS	high-level computer system simulator	高级计算机系统模拟器
HDA	head disk assembly	头盘组合件
HDAM	hierarchical direct access method	分层直接存取法
HDD	hard disk drive	硬磁盘驱动器
HDL	hardware description language	硬件描述语言
HDLC	high-level data link control (procedures)	高级数据链路控制(规程)
HDSL	high-bit-rate digital subscriber line	高比特率数字用户专用线
HDSS	holographic data storage system	全息数据存储系统
HEC	header error control	信头差错控制
HEMT	high electron mobility transistor	高电子迁移率晶体管
HFC	hybrid fiber coaxial cable	混合光纤同轴电缆
HFET	heterojunction field effect transistor	异质结场效应晶体管
HIDAM	hierarchical indexed direct access method	分层索引直接存取法
HIPO	hierarchical input, processing, output	层次输入–处理–输出(图)
HIPPI	high performance parallel interface	高性能并行接口
HISAM	hierarchical indexed sequential access method	分层索引顺序存取法
HLR	home location register	归属位置寄存器
HLS	hue-lightness-saturation	色调–亮度–饱和度(模型)
HMC	human-computer interaction and communication	人机交互和通信
HMD	head-mounted display	头盔显示器
HMM	hidden Markov model	隐式马尔可夫模型
HOLAP	hybrid online analysis processing	混合联机分析处理
HOOD	hierarchical object-oriented design	层次的面向对象设计
HP	Hewlett-Packard Co.	惠普公司,HP 公司
HPF	high pass filter	高通滤波器
HPF	high performance FORTRAN	高性能 FORTRAN(语言)
HPIB	Hewlett-Packard interface bus	惠普[公司]接口总线
HPSG	head-driven phrase structure grammar	中心词驱动短语文法
HQS	high quality screening	高质量挂网(技术)
HSAM	hierarchical sequential access method	分层顺序存取法
HSV	hue-saturation-value	色调–饱和度–明度值(模型)
HTML	hyper text markup language	超文本置标语言

缩略语	英文全名	中文译名
HTTP	hypertext transfer protocol	超文本传送协议
HyTime	Hypermedia Time-based Structuring Language	基于时间的超媒体结构化语言
IBE	ion beam etching	离子铣
IBG	interblock gap	[记录]数据块间隙
IBM	International Business Machines Corp.	国际商业机器公司,IBM 公司(美国)
IBM	image-based modeling	基于图像的造型
IBM-DOS	IBM Disk Operating System	IBM 公司磁盘操作系统
IBR	image-based rendering	基于图像的绘制
IC	integrated circuit	集成电路
ICA	independent component analysis	独立元分析
ICAI	intelligent computer-assisted instruction	智能计算机辅助教学
ICAM	integrated computer-aided manufacturing	综合计算机辅助制造
ICC	International Color Consortium	国际颜色联盟
ICCC	International Computer Communications Conference	国际计算机通信会议
ICMP	internet control message protocol	互连网控制消息协议
ICMS	integrated circuit message switch	集成电路信息交换设备
ICN	inteconnection network	互联网
ICN	information control network	信息控制网络(模型)
ICOT	Institute for New Generation Computer Technology	新一代计算机技术研究所(日本)
ICP	ion coupling	感应耦合等离子
ID	icon dictionary	[基本]图符字典
IDA	The Institute for Defense Analysis	(美国)国防部防御分析研究所
IDB	intensional database	内涵数据库
IDE	integrated data environment	数据集成环境
IDE	integrated drive electronics	集成驱动器电子线路(接口),IDE(接口)
IDE	interactive data entry	交互式数据录入
IDE	Interactive Development Environments	交互式开发环境(公司)
IDEA	international data encryption algorithm	国际数据加密算法
IDEF	integrated computer-aided manufacturing definition method	综合计算机辅助制造工程定义方法,ICAM 定义方法
IDFT	inverse discrete Fourier transform	离散傅里叶逆变换
IDL	interface definition language	接口定义语言
IDLC	integrated digital loop carrier	集成数字环载波
IDMS	integrated data base management system	综合数据库管理系统
IDN	integrated digital network	综合数字网

缩略语	英文全名	中文译名
IDS	information data system	信息数据系统
IDSS	intelligent decision support system	智能决策支持系统
IDT	interactive design tool	交互设计工具
IE	ion etching	等离子蚀刻
IEC	International Electrotechnical Commision	国际电工委员会
IEEE	Institute of Electrical and Electronics Engineers	电气和电子工程师学会(美国)
IETF	Internet Engineering Task Force	因特网工程任务部
IETM	interactive electronic technical manual	交互式电子技术手册
IFIP	International Federation for Information Processing	国际信息处理联合会
IFS	installable file system	可安装文件系统
IGBT	insulated gate bipolar transistor	绝缘栅双极晶体管
IGES	initial graphical exchange specification	初始图形交换规范
IGP	interior gateway protocol	内部网关协议
IGRP	interior gateway router protocol	内部网关路由协议
IGS	interactive graphics system	交互图形系统
IHL	Internet header length	因特网报头长度
II	initial interval	启动间距
IIL	integrated injection logic	集成注入逻辑电路
IKE	Internet key exchange	Internet 密钥交换
I^2L	integrated injection logic	集成注入逻辑[电路]
ILP	instruction level parallelism	指令级并行性
ILPP	instruction level parallel processing	指令级并行处理
IMP	interface message processor	接口[报文处理]机
IMS	information management system	信息管理系统
IN	intelligent network	智能网
INGRES	interactive graphic and retrieval system	交互式图示检索系统
INTERLISP	interactive LISP	交互 LISP 语言
IOD	input/output device	输入输出设备
I/O	input/output	输入输出
IP	internet protocol	网际协议
IPC	instructions per cycle	每个周期平均执行的指令数
IPD-CMM	capability maturity model for integrated product development	集成产品开发能力成熟度模型
IPG	information power grid	信息资源网格
IPI	intelligent peripherals interface	智能外围设备接口,IPI 接口
IPL	information processing language	信息处理语言

缩略语	英文全名	中文译名
IPSEC	IP security	(IETF 制定的)网际协议安全性
IPv6	Internet Protocol version 6.0	第 6.0 版网际协议
Ipv4	Internet Protocol version 4.0	第 4.0 版网际协议
IPX	internetwork packet exchange	网间数据包交换(协议),网际包交换(协议)
IQLA	International Quantitative Linguistics Association	国际计量语言学学会
IR	information retrieval	信息检索
IR	infrared radiation	红外
IR	instruction register	指令寄存器
μIR	microinstruction register	微指令寄存器
IRS	information retrieval system	信息检索系统
IS	intermediate system	中间系统
IS	international standard	国际标准
ISA	industry standard architecture	工业标准体系结构
ISAM	indexed sequential access method	索引顺序存取方法
iSCSI	Internet SCSI	因特网 SCSI 接口
ISDN	integrated services digital network	综合业务数字网
ISDOS	information system design and optimization system	信息系统设计和优化系统
ISG	inter-sector gap	扇区后间隙
IS-IS	intermediate system to intermediate system	IS-IS(协议)
ISO	International Standards Organization	国际标准化组织
ISO/OSI	International Organization for Standardization Open System Interconnection Model	ISO/OSI 模型,国际标准化组织-开放系统互连基准模型
ISP	Internet service provider	因特网服务提供者
ITSEC	Information Technology Security Evaluation Criteria	信息技术安全评估准则
ITT	International Telephone and Telegraph Corporation	国际电话电报公司(美国)
ITU	International Telecommunication Union	国际电信联盟
IVD	integrated voice and data	综合语音与数据
JCL	job control language	作业控制语言
JDBC	Java database connectivity	Java 数据库互连
JIS	Japanese Industrial Standard	日本工业标准
JIT	just in time	及时
JITP	just in time production	及时生产
JLD	joint-dependent local deformation	关节相关的局部变形
JOSS	Johnniac open-shop system	Johnniac 开放系统(语言)
JOVIAL	Jules own version of the international algorithmic language	国际算法语言的朱尔斯专用文本

缩略语	英文全名	中文译名
JPEG	Joint Photographic Expert Group	联合静止图像专家组
JPL	Jet Propulsion Laboratory	喷气推进实验室
JSD	Jackson system development	Jackson 系统开发[方法]
JSP	Jackson structured programming	Jackson 结构化程序设计
JTC	Joint Technical Committee	联合技术委员会
JTM	job transfer and manipulation	作业传送和操纵
JVM	Java virtual machine	Java 虚拟机
KADS	knowledge-aided document system	知识辅助文档系统
KAS	knowledge acquisition system	知识获取系统
KB	knowledge base	知识库
KBMS	knowledge base management system	知识库管理系统
KBS	knowledge-base system	知识库系统
KDC	key distribution center	密钥分发中心
KDD	knowledge discovery in database	数据库知识发现
KL	kernel language	核心语言
KQML	knowledge query and manipulation	知识查询与操纵语言
KRL	knowledge representation language	知识表示语言
KSR	keyboard send/receive	键盘发送接收
KWIC	keyword in context indexing	上下文中的关键字检索
KWIPS	kilo Whetstone instructions per second	千条 Whetstone 指令每秒
LAN	local area network	局域网
LAP	link access procedure	链路接入规程
LAP B	link access procedure-balanced	平衡型链路接入规程
LAP D	link access control procedure on the D-channel	D 通道链路接入规程
LAP F	Link Access Procedure for Frame-mode hearer services	用于帧中继的链路访问规程
LAT	local area transport	局部区域运输(协议)
LBA	linear bounded automaton	线性有界自动机
LBA	logic block addressing	逻辑块编址(模式)
LCD	liquid crystal display	液晶显示,液晶显示器
LCF	logic for computable functions	可算函数逻辑
LCF	low cost fiber	低成本光纤
LCP	link control protocol	链路控制协议
LCP	logical construction of programs	程序[的]逻辑构造,程序的逻辑结构(方法)
LCS	logical construction of systems	系统[的]逻辑构造
LDAP	light weight directory access protocol	轻量级目录服务协议

缩略语	英文全名	中文译名
LDD	lightly doped drain	轻搀杂漏区
LDL	logical data base level	逻辑数据库级
LED	light-emitting diode	发光二极管
LEO	low earth orbit	近地轨道
LFA	local feature analysis	局部特征分析
LFG	lexical functional grammar	词汇功能语法
LFSR	linear feedback serial register	线性反馈串行寄存器
LFU	least frequently used	最不常用
LGMR	laser guidance magnetic recording	激光制导磁记录
LinMOS	linear CMOS	线性 CMOS
LIP	loop initialize process	环初始化进程
LISP	List Processing	表处理语言,LISP 语言
LIW	long instruction word	长指令字
LLC	logical link control	逻辑链路控制
LMDS	local multipoint distribution services	本地多点分配业务
LMSC	LAN/MAN Standards Committee	局域网和城域网标准委员会
LOS	logical output structure	逻辑输出结构
LP	low pass	低通(滤波器)
LPC	linear prediction coding	线性预测编码
LPC	linear predictive coefficient	线性预测系数
LPC	local procedure call	本地过程调用
LPCVD	low pressure chemical vapor deposition	低压化学气相淀积
LPF	low pass filter	低通滤波器
LPS	logical processing structure	逻辑处理结构
LPSL	low press field effect transistor	低夹断场效应晶体管
LR	location register	定位寄存器
LRC	longitudinal redundancy check	水平冗余检验,纵向冗余检验
LRU	least recently used	最近最少使用
LSAP	link services access point	链路服务访问点
LSB	least significant bit	最低有效位
LSI	large scale integration	大规模集成[电路]
LSP	link state packet	链路状态分组
LSSD	level sensitive scan design	电平敏感扫描设计
LTO	linear tape open	线性磁带(系统)开放(技术)
LU	line unit	线路单元
LUT	look-up table	查找表

缩略语	英文全名	中文译名
LVD	laser video disc	激光视盘
LWIRD	long-wave length infrared detector	长波长红外探测器
MAC	medium access control	介质访问控制
MAHT	machine-aided human translation	机助人译
MAI	multiple access interference	多址接入干扰
MANDATE	multiline automatic network diagnostic and transmission equipment	多线路自动网络诊断和传输设备
MAN	metropolitan area network	城域网
MAP	manufacturing automation protocol	制造自动化协议
MAP	mobile application part	移动应用部分
MAR	memory address register	存储器地址寄存器
MARGIE	meaning analysis, response generation and inference on English	英语的语义分析、反应生成和推断(系统)
MathML	Mathematical Markup Language	数学置标语言
MAU	medium access unit	介质访问单元
MAVICA	magnetic video camera	磁性视频摄像
MB	model base	模型库
MBE	molecular beam epitaxy	分子束外延
MBMS	model base management system	模型库管理系统
MBR	memory buffer register	存储器缓冲寄存器
MCA	micro channel architecture	微通道体系结构
MCC	Micro-electronics and Computer Technology Corp.	微电子与计算机技术公司(美国)
MCC	Multiple Computer Complex	多计算机联合体(美国)
MCI	Microwave Communications, Inc.	微波通信公司(美国)
MCM	multichip module	多芯片模块
MCPC	multi-channel per carrier	单载波多路
MCS	microcomputer system	微型计算机系统
MCU	machine control unit	机器控制单元
MCU	microcontrol unit	微控制器
MCU	micro-programmed control unit	微程序控制器
MDA	model-driven architecture	模型驱动的体系结构
MDA	monochrome display adapter	单色显示适配器
MDI	medium dependent interface	介质相关接口
MEO	middle earth orbit	中间地球轨道
MESFET	metal-semiconductor field effect transistor	金属-半导体场效应晶体管
MF	more fragment	尚有分段

缩略语	英文全名	中文译名
MFLOPS	million floating point operations per second	百万[次]浮点运算每秒
MFM	modified frequency modulation	改进调频(制)
MHEG	Multimedia and Hypermedia Expert Group	多媒体和超媒体专家组
MHS	message handling system	消息处理系统
MIB	management information base	管理信息库
MIC	memory in cassette	磁带盒内存储器(技术)
MIDAS	modified integration digital to analog simulator	改进的集成数模仿真器
MIDI	musical instrument digital interface	乐器数字接口
MIG	metal-in-gap	隙含金属(磁头)
MIMD	multiple-instruction stream multiple-data stream	多指令流多数据流
MIME	multipurpose Internet mail extension	多用途因特网电子邮件扩展(协议)
MIPS	million instructions per second	百万[条]指令每秒
MISD	multiple-instruction stream single-data stream	多指令流单数据流
MIS	management information services	管理信息服务
MIS	manufacturing information system	制造信息系统
MIT	management information tree	管理信息树
MIT	Massachussetts Institute of Technology	麻省理工学院(美国)
ML	macrolanguage	宏加工语言
ML	metalanguage	元语言
MM	main memory	主存储器
MMCD	multimedia CD	多媒体数字光盘
MMDS	multichannel multipoint distribution services	多路多点分配业务
MMIC	monolithic microwave integrated circuit	单片微波集成电路
MMS	manufacturing message specification	制造报文规范
MMU	main memory unit	主存储器
MMU	memory management unit	存储器管理部件,存储管理部件
MO	managed object	被管对象
MOCVD	metal organism CVD	金属有机物化学汽相淀积法
MOD	moving object database	移动对象数据库
MODFET	modulation doped field effect transistor	调制掺杂场效应晶体管
MOLAP	multidimentional online analysis processing	多维联机分析处理
MOPS	micromonitor operating system	微型监控器操作系统
MOS	metal-oxide-semiconductor	金属-氧化物-半导体
MOSFET	metal-oxide-semiconductor field effect transistor	金属-氧化物-半导体场效应晶体管
MOTIS	message oriented text interchange system	面向消息的正文交换系统

缩略语	英文全名	中文译名
MP	multimedia processor	多媒体处理器
MP	multiprocessor	多处理机
MPC	multimedia personal computer	多媒体个人计算机
MPEG	Moving Picture Expert Group	运动图像专家组
MPI	message passing interface	消息传递接口
MPP	massively parallel processing	大规模并行处理
MQ	message queuing	报文队列
MR	magnetic resistive	磁变阻(磁头)
MRAM	magnetic-resistive RAM	磁电阻随机存取存储器
MRD	machine readable dictionary	机器可读词典
MRP	manufacturing resource planning	制造资源规划
MRP	material requirement planning	物料需求计划
MRU	maximum receive unit	最大接收单元
MS	message store	消息存储[单元]
MS	mobile station	移动台
MSB	most significant bit	最高有效位
MSC	moving switcher	移动交换机
MS-DOS	Microsoft Disk Operating System	微软公司磁盘操作系统
MSE	mean square error	均方误差
MSI	medium scale integration	中规模集成[电路]
MSIPS	million synchronous instructions per second	百万条同步指令每秒
MSN	Manhattan street network	曼哈顿街道网
MSS	MAN switching system	城域网交换系统
MSS	mass storage system	海量存储系统
MSS	multispectral scanner	多光谱扫描仪
MT	machine translation	机器翻译
MTA	message transfer agent	消息传送代理
MTBF	mean time between failures	平均故障间隔时间
MTL	message transfer sublayer	消息传送子层
MTOPS	millions of theoretical operations per second	百万次理论运算每秒
MTS	message transfer system	消息传送系统
MTTF	mean time to failure	平均无故障时间
MTTR	mean time to repair	平均修复时间
MTU	maximum transfer unit	(网络)最大传输单元
MTX	mobile telephone exchange	移动电话交换局

缩略语	英文全名	中文译名
MVL	multivalued logic	多值逻辑电路
MVS	multiple virtual storage (operating system)	多虚拟存储器(操作系统)
MWIPS	million Whetstone instructions per second	百万条 Whetstone 指令每秒
MWM	MOTIF window manage	MOTIF 窗口管理器
NAP	network access process	网络存取进程
NAS	network attached storage	附网存储
NAS	National Academy of Sciences	(美国)国家科学院
NASA	National Aeronautics and Space Administration	(美国)国家宇航局
NAT	network address translation	网络地址翻译
NATO	North Atlantic Treaty Organization	北大西洋公约组织
NAV	network assigned vector	网络配给向量
NBM	narrow band modulation	窄带调制(码)
NBS	National Bureau of Standards	国家标准局(美国)
NC	network computer	网络计算机
NC	numerical control	数值控制,数控
NCC	network control center	网络控制中心
NCP	network control protocol	网络控制协议
NCSA	National Computational Science Alliance	(美国)国家计算科学同盟会
NCSC	National Computer Security Center	(美国)国家计算机安全中心
NDBMS	network data base management system	网络数据库管理系统
NDC	normalized device coordinate system	规格化设备坐标系
NDD	network data dictionary	网络数据字典
NDS	domain name system	域名系统
NE	network element	网络元素
NEBULA	natural electronic business users language	电子商业用户自然语言
NEC	Nippon Electric Co.	日本电气公司,NEC 公司
NELIAC	Navy Electronics Laboratory International ALGOL Compiler	海军电子实验室国际 ALGOL 编译程序
NFA	nondeterministic finite automaton	非确定型有限自动机
NFS	Network File System	网络文件系统(美国 Sun 公司)
1NF	the first normal form	第一范式
2NF	the second normal form	第二范式
3NF	the third normal form	第三范式
4NF	the fourth normal form	第四范式
NGI	next generation Internet	下一代因特网
NIC	network information center	网络信息中心

缩略语	英文全名	中文译名
N-ISDN	narrowband integrated services digital network	窄带综合业务数字网
NIST	National Institute of Standards and Technology	国家标准化技术研究所(美国)
NLN	neural logic network	神经逻辑网络
NLP	natural language processing	自然语言处理
NLSP	Netware Link Service Protocol	Netware 链路服务协议
NLU	natural language understanding	自然语言理解
NMC	network management center	网[络]管[理]中心
NMOS	N-channel metal-oxide-semiconductor	N 沟道金属-氧化物-半导体
NMR	nuclear magnetic resonance	核磁共振
NMS	network management station	网络管理站
NMT	network management terminal	网络管理终端
NNI	network-network interface	网间接口
Non-GEO	non-geostationary earth orbit	对地静止轨道
NOS	network operating system	网络操作系统
NOW	network of workstation	工作站网
NP	noun phrase	名词短语
NPACI	National Partnership for Advanced Computational Infrastructure	国家高级计算设施联合会(美国)
NPL	nonprocedural language	非过程语言
NPR	non-photorealistic rendering	非真实感图形绘制
NPT	non-packet terminal	非分组式终端,非包式终端
NRC	National Research Council	国家科学研究委员会(加拿大)
NRM	normal response mode	正常响应模式
NRZ1	non-return-to-zero change on one	逢 1 变化不归零制(码)
NRZ	non-return-to-zero	不归零制(记录格式)
NSA	National Security Agency	国家安全局(美国)
NSF	National Science Foundation	国家科学基金会(美国)
NSS	network and switching subsystem	网络和交换系统
NURBS	non-uniform rational B-spline	非均匀有理 B 样条
NV RAM	non-volatile RAM	非易失性随机存储器
OA	office automation	办公自动化
OAS	open application system	开放应用系统
OBE	office by example	实例办公语言
OC-48	optical carrier, level 48	光载波信号线路 48 级
OC-12	optical carrier, level 12	光载波信号线路 12 级
OC-3	optical carrier, level 3	光载波信号线路 3 级

缩略语	英文全名	中文译名
OCR	optical character reader	光学字符阅读机
OCR	optical character recognition	光学字符识别
ODBC	open database connection	开放式数据库连接
ODBC	open database connectivity	开放式数据库互连
ODP	open distributed processing	开放式分布处理
OEIC	opto-electronic integrated circuit	光电集成电路
OEM	object exchange model	对象交换模型
OEM	original equipment manufacturer	初始设备制造厂
OGSA	open grid services architecture	开放式网格服务体系结构
OIS	office information system	办公信息系统
OLAP	online analysis processing	联机分析处理
OLE	object linking and embedding	对象链接与嵌入(功能)
OLTP	on-line transaction processing	联机事务处理
OLTP	on-line transaction processing system	联机事务处理系统
OMC-R	operation and maintenance center-radio	操作与维护中心-无线电
OMC-S	operation and maintenance center-switching	操作与维护中心-交换机
OMG	Object Management Group	对象管理组织
OMG-IDL	OMG Interface Definition Language	OMG 组织提出的接口定义语言
OMR	optical mark reader	光学标记阅读机
OMT	object modeling technique	对象模型化技术
ONU	optical network unit	光纤网部件
OO	object-oriented development method	OO 方法,面向对象的开发方法
OOA	object-oriented analysis	面向对象的分析
OODBMS	object-oriented database management system	面向对象数据库管理系统
OOD	object-oriented design	面向对象的设计
OOI	object-oriented implementation	面向对象的实现
OOP	object-oriented programming	面向对象的程序设计
OOSD	object-oriented structured design	面向对象的结构化设计
OPS	operations per second	运算(次数)每秒
OR	operation ratio	运转率,运行率
ORB	object request broker	对象请求代理
ORDBMS	object-relational database management system	对象-关系数据库管理系统
OS	open system	开放系统
OS	operation system	运营系统
OSF	Open Software Foundation	开放软件基金会

缩略语	英文全名	中文译名
OSI	open system interconnection	开放系统互连
OSIE	open system interconnection environment	开放系统互连环境
OSI/RM	open system interconnection reference model	开放系统互连基准(参考)模型
OSPF	open shortest path first	开放式最短通路优先(算法)
Oss	optical carriers	光纤载体
OTM	office task management	办公任务管理(模型)
OTP EPROM	one time programmable EPROM	一次可编程 EPROM
OUM	ovonic unified memory	相变存储器
PABX	private automatic branch exchange	专用自动小交换机
PAC	probably approximately correct	概率近似正确(学习模型)
PAD	packet assembler/disassembler	分组装拆器,包装拆器
PAD	problem analysis diagram	问题分析图
PAL	programmable array logic	可编程阵列逻辑[电路]
PAM	pulse amplitude modulation	脉[冲]幅[度]调制
PAP	password authentication protocol	口令鉴别协议
PARC	Palo Alto Research Center	Palo Alto 研究中心(美国 Xerox 公司)
PARLOG	Parallel Programming In Logic	并行逻辑程序设计(语言),PARLOG(语言)
PASS	program alternative simulation system	程序交替模拟系统
PAN	personal area network	私人网
PBX	private branch exchange	专用小交换机
PC	personal computer	个人计算机,PC 机
PC	programmable controller	可编程控制器
PC	program counter	程序计数器
PCA	principal component analysis	主成分分析
PCB	printed circuit board	印制[电路]板
PC-DOS	Personal Computer-Disk Operating System	个人计算机磁盘操作系统
PCF FORTRAN	parallel context-free FORTRAN	并行上下文无关 FORTRAN(语言)
PCFG	probabilistic context-free grammar	概率型上下文无关文法
PCI	peripheral component interconnection	外围部件互连
PCI	protocol control information	协议控制信息
PCI,PCI-X	Peripheral Connection Interface	外围部件互连(接口)
PCL	printer control language	打印机控制语言
PCMCIA	Personal Computer Memory Card International Association	个人计算机存储卡国际协会
μPC	microprogram counter	微程序计数器
PCM	pulse code modulation	脉[冲编]码调制

缩略语	英文全名	中文译名
PCN	personal communication network	个人通信网
PCS	Personal Conferencing Specification	个人会议标准(规范)
PCT	personal construct theory	个人构造理论
PCTE	portable common tool environment	可移植的公共工具环境
PCWG	Personal Conferencing Work Group	个人会议工作组
PD	phase detector	鉴相器
PDA	personal digital assistant	个人数字助理
PDA	pushdown automaton	下推自动机
PDAU	physical delivery access unit	物理投递访问单元
PDDI	fiber distributed data interface	光纤分布式接口
PDES	product data exchange specification	产品数据交换标准
PDF	portable document format	可移植表示格式(语言)
PDL	program description language	程序描述语言
PDL	program design language	设计[性]程序语言
PDM	product data management	产品数据管理
PDN	public data network	公用数据网
PDP	parallel distributed processing	并行分布式处理
PDR	processing data rate	数据处理速率
PDS	premises distribution system	宅院布线系统
PDU	packet data unit	分组数据单元
PDU	protocol data unit	协议数据单元
People-CMM	Capability Maturity Model for People	人员能力成熟度模型
PE	phase encoding	调相制,相位编码
PE	plasma etching	等离子蚀刻
PE	processing element	处理部件
PERT	program evaluation and review technique	计划评价与评审技术
PGP	Pretty Good Privacy	良好隐私(事实上的全球电子邮件加密标准)
PHB	per-hop-behavior	每跳行为
PHEMT	psedo-morphic high electro mobility transistor	伪晶高电子迁移率晶体管
PHIGS	programmer's hierachical interactive graphics system	PHIGS 图形标准
PHIGS	programmer's hierarchical interactive graphics standard	程序员分层交互图形标准
PHY	physical sublayer	物理子层
PI	process interactive	进程交互(法)
PIC	programmable intelligent computer	可编程智能计算机
PICS	production information and control system	生产信息与管理系统

缩略语	英文全名	中文译名
PICS	protocol implementation conformance statement	协议实现一致性声明
PID	process identifier	进程标识符
PID	proportional-integral-differential	比例积分微分
PID	proportional-integral-differential controller	比例积分微分控制器,PID 控制器
PIM	parallel inference machine	并行推理机
PIM	processor-in-memory	处理器在内存
PIMOS	parallel inference machine operating system	并行推理机操作系统
PIN	personal identification number	个人认证号
PIO	programmed input/output	程序控制输入输出
PKI	public key infrastructure	公开密钥设施
PLA	programmable logic array	可编程逻辑阵列
PLATO	programmed logic for automatic teaching operations	自动教学业务用程序逻辑
PLATO	programming language for automatic teaching operations	自动教学业务用的程序设计语言
PLD	programmable logic device	可编程逻辑器件
PLL	phase locked logic	锁相逻辑[电路]
PLL	phase-locked loop	锁相环
PLM	programmable logic machine	可编程序逻辑机
PL/1	Programming Language/1	程序设计语言 1,PL/1 语言
PLS	physical signaling	物理信令(号)
PLTL	propositional linear temporal logic	命题线性时态逻辑
PM	phase modulation	调相(制)
PM	physical medidum	物理介质
PM	physical medium	(ATM)物理介质(子层)
PMA	physical medium attachment	物理介质连接
3PM	3 phase modulation	3 单元调相制
3PM	3 position modulation	3 单元调制(码)
PM	progressive meshes	递进网格
PMD	physical medium-independent sublayer of local area network	局域网物理介质无关子层
PMOS	P-channel MOS	P 沟道 MOS
PNNI	private network to network interface	专用网络间接口
PNN	probabilistic neural network	概率神经网络
PON	passive optical network	无源光纤网
POS	personal operating space	私人工作空间
POS	point of sale terminal	销售点终端
POSIX	portable operating system interface for computer environments	计算机环境的可移植操作系统接口

缩略语	英文全名	中文译名
POSIX	Portable Operating System Interface(UNIX-like)	UNIX 可移植操作系统界面(标准)
POSIX	portable operating system UNIX	可移植的 UNIX 操作系统
POSLA	pitch synchronous overlap add	基音同步叠加
PP	preposition phrase	介词短语
P2P	person to person	个人到个人
PPA	probabilistic pushdown automaton	概率下推自动机
PPP	point-to-point protocol	点对点连接协议
PRAM	parallel random access machine	并行随机存取机
PRML	partial response maximum likelihood	部分响应最大似然(信号处理技术)
PROLOG	Programming in Logic	逻辑程序设计(语言),PROLOG(语言)
P2ROM	product-PROM	产品-可编程只读存储器
PROM	programmable read-only memory	可编程只读存储器
PRPS	phase rectifying power supply	多相整流电源
PSA	problem statement analyzer	问题陈述分析程序
PSDN	packet swiched data network	分组交换数据网
PSG	phrase structure grammar	短语结构语法
PSI	personal sequential inference machine	个人顺序推理机
PSK	phase shift keying	相移键控
PSL	problem statement language	问题陈述语言
PSM	problem solving description mechanism	问题求解描述机制
PSOLA	pitch-synchronous overlap add	音调同步重叠相加法
PSOS	plug-in silicon operating system	插入式硅操作系统
PSP	personal software process	个人软件过程
PSTN	public switched telephone network	公用交换电话网
PTE	page table entry	页表项
P to P	person to person	个人到个人
PT	payload type	净荷类型
PUP	PARC universal packet	Palo Alto 研究中心通用分组
PVC	permanent virtual circuit	永久性虚电路
PVM	parallel virtual machine	并行虚拟机
PVP	parallel vectors processors	并行向量处理(机)
QAM	quadratic amplitude modulation	正交调幅
QBE	Query By Example	实例查询语言(美国 IBM 公司)
Qb	quantum bit	量子位
Qbit	quantum bit	量子位

缩略语	英文全名	中文译名
QCIF	quarter-CIF	四分之一 CIF 图像格式
QIC	quarter inch tape	1/4 英寸盒带
QoS	quality of service	服务质量
QR	quick response	快速响应
QUEL	query language	查询语言
RA	ripple adder	行波加法器
RAID	redundant array of inexpensive disk	廉价冗余磁盘阵列
RAM	random access machine	随机存取机
RAM	random access memory	随机存取存储器
RAS	reliability, availability and seviceability	可靠性、可用性及可维[修]性
RB	route bit	路由比特
R/C	return/count	返回-计数(寄存器)
RDA	remote database access	远程数据库访问
RDBMS	relational database management system	关系数据库管理系统
RDRAM	Ramdus dynamic random access memory	Ramdus 动态随机存储器
RF	radio frequency	射频
RFC	Request for Comments	征求意见文件
RFC2210	Request for Comments 2210	第 2210 号征求意见文件
RG	regular grammar	正则文法,3 型文法
RGB	red-green-blue	红-绿-蓝(模型)
RIE	reactive ion etching	反应离子蚀刻
RIG	Reuse Library Interoperability Group	复用库互操作组织
RIP	raster image processor	光栅图像处理器,栅格图像处理器
RIP	routing information protocol	路由信息协议
RISC	reduced instruction set computer	精简指令集计算机
RLLC	run-length-limited code	游程长度受限码
rlogin	remote login	远程登录
RML	requirements modeling language	需求模型化语言
RMON MIB	remote network monitoring management information base	远程网络监控管理信息库
R-M	reed muller	舌簧混沙机
RMON	remote network monitorying	远程网络监控
ROC	receiver operating cure	ROC 曲线
ROLAP	relational online analysis processing	关系联机分析处理
ROM	read-only memory	只读存储器
ROSE	remote operations service element	远程操作服务要素
RPC	remote procedure call	远程过程调用

缩略语	英文全名	中文译名
RPG	report program generator	报表程序生成程序
RPLA	reprogrammable logic array	可重编程逻辑阵列
R-S	Reed-Solomon (code)	里德-索罗门(码)
RS	remote sensing	遥感
RSA	Rivest-Shamir-Adleman (method)	RSA 方法(一种数据安全编码方法)
RSEXEC	resource-sharing executive	资源共享执行程序
RSL	requirements statement language	需求陈述语言
RS-PC	reed-solomon product code	里德-索罗门产品编码
RSVP	network resource reservation protocol	网络资源预约协议
RTA	rapid thermal anneal	快速退火(技术)
RTC	real time clock	实时时钟
RTD	resonant tunneling device	共振隧穿器件
RTL	register transfer language	寄存器传送语言
RTL	resistor transistor logic	电阻-晶体管逻辑
RTP	Real-time Transfer Protocol	实时传输协议
RTS/CTS	request to send/clear to send	请求发送与清除发送
Rt-VBR	real-time variable bit rate	实时可变比特率(服务)
RWC	real world computing	真实世界计算
SA	structured analysis	结构化分析
SA	source address	源地址
SA	security association	安全关联
SAA	system application architecture	系统应用体系结构
SA-CMM	capability maturity model for software acquisition	软件获取能力成熟度模型
SADT	structured analysis and design technique	结构化分析与设计技术
SAG	SQL Access Group	SQL 访问组
SAGE	semi-automatic ground environment computer	半自动地面防空警戒计算机,赛其防空警戒计算机
SALT	symbolic algebraic language translator	符号代数语言翻译程序
SAMM	systematic activity modeling method	系统化活动建模方法
SAM	sequential access memory	串行存取存储器
SAM	serial access memory	串行存取存储器
SAN	storage area network	存储区域网
SAP	service access point	服务访问点
SAR	segmentation and reassembly	分段与重组
SAS	serial attached SCSI	串行嵌入式 SCSI 接口
SASE	specific application service element	特定应用服务要素

缩略语	英文全名	中文译名
SATA	serial ATA	串行 ATA 接口
SAX	simple API for XML	XML 的简单应用编程接口
SBI	serial bus interface	串行总线接口
SC	structure chart	结构图
SC	Subcommittee	分委员会(国际标准化组织)
SCCS	source code control system	源[代]码控制系统
SCFL	source coupled FET logic	源耦合场效应晶体管逻辑电路
SCOOP	system for computerization of office processes	计算机化办公事务处理系统
SCR	silicon controlled rectifier	可控硅整流器
SCS	structured cabling system	结构化布线系统
SCSI	small computer system interface	小计算机系统接口,SCSI 接口
SD	structured design	结构化设计
SD	super high desity digital video disc	超高密度数字视盘
SDD	system for distributed databases	分布式数据库系统
SDE	shared data environment	数据共享环境
SDE	software development environment	软件开发环境
SDFL	Schottky diode FET logic	肖特基二极管场效应晶体管逻辑电路
SDF-4	stroke density features in four direction	四方向笔画密度特征
SDH	synchronous digital hierarchy	同步数字体系
SDL	structural descriptive language	结构描述语言
SDLC	synchronous data link control	同步数据链路控制(规程)
SDLT	super digital linear tape	超级数字线性磁带(技术)
SDR	single data rate	单数据速率
SDRAM	synchronous DRAM	同步动态随机存储器(芯片)
SDRC	Structural Dynamics Research Corp.	结构动态研究公司(美国)
SDS	space division switching	空分交换机
SDTS	syntax-directed translation schema	语法导向翻译模式
SDU	service data unit	服务数据单元
SE	software engineering	软件工程
SE-CMM	capability maturity model for system engineering	系统工程能力成熟度模型
SEE	software engineering environment	软件工程环境
SEED	self-electrooptic effect device	自电光效应器件
SEQUEL	structured English query language	结构化的英语查询语言
SET	Standard d'Echange et de transfer	(法国和欧洲)工程数据交换标准
SETL	Set-Oriented Language	面向集合的语言,SETL 语言

缩略语	英文全名	中文译名
SGLDM	spatial grey-level dependence matrix	空间灰度关系矩阵
SGML	standard generalized markup language	标准通用置标语言
SIG	special interest group	专项组,专业组
SIGOA	Special Interest Group on Office Automation	办公自动化专业组(美国计算机协会)
SIGOIS	Special Interest Group on Office Information System	办公信息系统专业组(美国计算机协会)
SILS	standard for interoperable local area network security	互操作局域网的安全标准
SIM	subscribe identity module	用户身份组件
SIMD	single-instruction stream multiple-data stream	单指令流多数据流
SIMULA	simulation language	SIMULA[仿真]语言
SIP	SMDS interface protocol	SMDS 接口协议
SISD	single-instruction stream single-data stream	单指令流单数据流
SL	specification language	规约语言
SLAM	simulation language for analogue modeling	模拟建模的仿真语言
SLIP	serial line internet protocol	串行线路网际协议
SLM	space laser modulator	空间光调制器
SM	semiconductor memory	半导体存储器
SMAP	system management application protocol	系统管理应用协议
SMART	system for the mechanical analysis and retrieval of text	SMART 系统
SMD	storage module drive	存储模块驱动器(接口),SMD 接口
SMD	surface mounted device	表面安装器件
SMDS	switched multimegabit data service	交换式多兆位数据业务
SME	Society of Manufacturing Engineer	制造工程师协会(美国)
SMF	single mode fiber	单模光纤
SMI	structure of management information	管理信息结构
SMI	system management interrupt	系统管理中断
SMIL	Synchronized Multimedia Integration Language	同步多媒体集成语言
SML	standard metalanguage	标准元语言
SML	system message language	系统消息语言
SMM	system management mode	系统管理模式
SMPS	switching mode power supply	开关电源
SMP	symmetric multiprocessor	对称式多处理机
SMT	station management	(FDDI)站管理
SMT	surface mounting technology	表面安装技术
SMTP	Simple Mail Transfer Protocol	简单邮件传送协议

缩略语	英文全名	中文译名
SNA	system network architecture	系统网络体系结构
SNI	subscriber network interface	用户网络接口
SNIA	Storage Networking Industry Association	存储网络产业协会
SNMP	Simple Network Management Protocol	简单网络管理协议
SNOBOL	String-Oriented Symbolic Language	面向串的符号语言,SNOBOL 语言
SNRZ	synchronized non-return-to-zero	同步不归零(制)
SOC	system on a chip	片上系统
SOI	silicon-on-insulator	绝缘体上硅,单晶硅薄膜
SON	synchronous optical network	同步光纤网
SONET	synchronous optical network	同步光纤网
SOPC	system on programmable chip	可编程的片上系统
SP	stack pointer	栈指针
SP	structured programming	结构化程序设计
SPARC	scalable processor architecture	可缩放处理机体系结构
SPARC	Standards Planning and Requirements Committee (ANSI)	规划与需求标准委员会(美国国家标准学会)
SPC	set-point control	设定值控制
SPC	stored-program control	存储程序控制
SPEC	System Performance Evaluation Cooperative	系统性能评价合作组织
SPI	serial peripheral interface	串行外围接口
SPICE	Simulation Program With Integrated Circuit Emphasis	(程序名)
SPICE	Software Process Improvement and Capacity determine	软件过程改进和能力确定(项目)
SPLICE	Simulation Program With Large Scale Integrated Circuit Emphasis	(程序名)
SPMD	single program over multiple data streams	单程序多数据流
SPOOF	structure and parity-observing output function	结构与奇偶性观察输出功能(方法),SPOOF 法
SPOOL	simultaneous peripheral operations online	假脱机[操作]
SQL	structured query language	结构化查询语言
SRAM	static random access memory	静态随机存储器
SRB	source-routebridging	源路由选择网桥
SRI	Stanford Research Institute	斯坦福研究所(美国)
SRS	source-routeswitching	源路由选择交换
SRT	source-routetransparent	源路由透明(网桥)
SS	self-scheduling	自调度
SS	signalling system	信令系统

缩略语	英文全名	中文译名
SSA	serial storage architecture	串行存储体系结构
SSA	structured system analysis	结构化系统分析
SSAP	source services access point	源服务访问点
SSD	solid state disc	固态盘
SSI	small scale integration	小规模集成[电路]
SSL	secure sockets layer	安全套接字协议层
SSRAM	synchronous SRAM	同步静态随机存取存储器(芯片)
SS-TDMA	satellite switched time-division multiple access	卫星交换的时分多址
STARS	software technology for adaptable reliable system	自适应可靠系统的软件技术
STDBUS	standard bus	标准总线
STDM	statistic time division multiplexing	统计时分多路复用
STE	signalling terminal equipment	信号传输终端设备
STEP	standard for the exchange of product model data	产品模型数据交换标准
STMs	synchronous transport modules	同步传输模块
STM	synchronous transfer mode	同步传送模式
STP	shielded twisted pair	屏蔽双绞线
STRADIS	structured analysis, design and implementation of information systems	信息系统结构化分析、设计及实现
STSs	synchronous transport signals	同步传输信号
SVC	switched virtual circuit	交换[型]虚电路
SVCD	super video CD	超级 VCD 视盘
SVG	scalable vector graphic	可缩放向量图形
SVID	System V Interface Definitions	(UNIX)系统 V 界面(标准)
SVM	support vector machine	支持向量机
SW-CMM	capability maturity model for software	软件能力成熟度模型
TAB	tape automated bounding	载带自动焊
TAPI	telephone application programming interface	电话应用编程接口
TAP	TCP/IP access procedure	TCP/IP 接入规程
TBM	terabit memory	太位存储器
TC	technical committee	技术委员会
TC	terminal controller	终端控制器
TC	transmission convergence	传输会聚
TCB	trusted computing base	可信计算基
TCM	thermal conduction module	导热模块
TCM	trellis coded modulation	网格编码调制
TCP	Transmission Control Protocol	传输控制协议

缩略语	英文全名	中文译名
TCSEC	trusted computer system evaluation criteria	可信计算机系统评估准则
TCSP	theory of CSP	通信顺序进程理论
TDI	trusted database interpretation of the TCSEC	可信数据库指南
TDMA	time division multiple access	时分多址接入
TDM	time division multiplexing	时分多路复用
TDNN	time delay neural network	时间延迟神经网络
TDR	time-domain reflectometer	时域反射仪
TDS	time division switching	时分交换机
TE	terminal equipment	终端设备
TEMPEST	technique of electro-mechanical protection against emission and spurious transmission	防信息泄漏技术
TFT	thin film transistor	薄膜晶体管
TFTP	trivial file transfer protocol	普通文件传送协议
TLB	translation lookaside buffer	转换检测缓冲器
TLS	transport layer security	运输层安全(协议)
TMN	Telecommunication Management Network	电信管理网络
TMR	triple modular redundancy	三模冗余
TMS	truth maintenance system	真值维护系统
TNI	trusted network interpretation of the TCSEC	可信网络指南
TOP	technical and office protocol	技术和办公协议
TP	theoretical peak	理论峰值
TPC	Transaction Processing Council	事务处理委员会
TPDDI	twisted pair distributed data interface	双绞线分布式数据接口
TPS	transactions per second	事务处理(数)每秒
TRC	transversal redundancy check	纵横冗余检验,行列冗余检验
TRS	text retrieval system	正文检索系统
TSP	throng software process	群组软件过程
TSS	trapezoid self-scheduling	梯形[自]调度
TST	time-space-time	时间槽交换
TTCN	tree and tabular combined notation	协议测试用形式描述语言
TTL	transistor transistor logic	晶体管晶体管逻辑[电路]
TTS	text to speech	文语转换
UA	user agent	用户代理
UAL	user agent sublayer	用户代理子层
UBR	unspecified bit rate	未指定比特率(服务)
UBS	uninterruptable battery system	不间断电池系统
UCR	undercolor removel	基色去除函数

缩略语	英文全名	中文译名
UCS	Universal Multiple-Octet Coded Character Set	通用多八位编码字符集
UDM	uniform data model	统一数据模型
UDP	User Datagram Protocol	用户数据报协议
UGIS	urban geographic information system	城市地理信息系统
UI	UNIX International	UNIX 国际
UIDE	user interface development environment	用户界面开发环境
UIDT	user interface development tool	用户界面开发工具
UIL	user interface language	用户界面语言
UIMS	user interface management system	用户界面管理系统
UIMT	user interface management tool	用户界面管理工具
UIS	user interface system	用户界面系统
UITB	user interface tool box	用户界面工具箱
UITK	user interface toolkit	用户界面工具箱
ULSI	ultralarge scale integration	特大规模集成[电路]
UMA	uniform memory access	均匀存储器访问
UNC	University of North Carolina	北卡大学
UN/EDIFACT	United Nations/electronic data interchange for administration, commerce and transport	联合国用于行政、商业和运输的电子数据交换(标准)，联合国 EDIFACT(标准)
UNI	ATM network node interface	ATM 网间接口
UNI	user-network interface	用户-网络接口
UPC	universal product code	通用产品代码
UPS	uninterruptable power system	不间断电源
URI	uniform resource identifier	统一资源定位标记
URL	universal resource locator	统一资源定位器
USB	universal serial bus	通用串行总线
UTF	UCS transformation format	UCS 变形显现形式
UTM	universal transverse Mercator	UTM(坐标系统)
UTP	unshielded twisted pair	无屏蔽双绞线
UVE-PROM	ultraviolet EPROM	紫外线可擦可编程只读存储器，紫外线 EPROM
VAG	visual attribute grammar	可视属性文法
VAL	a value-oriented algorithmic language	面向值的算法语言
VAL	value algorithmic language	单赋值算法语言
VC	virtual call	虚呼叫
VC	virtual channel	虚通道
VC	virtual circuit	虚电路

缩略语	英文全名	中文译名
VCD	video CD	VCD 视盘
VCI	virtual channel identifier	虚通道标识符
VCO	voltage controlled oscillator	压控振荡器
VDC	virtual device coordinate space	虚拟的设备坐标空间
VDL	Vienna defination language	维也纳定义语言
VDM	Vienna development method	维也纳开发方法
VESA	Video Electronic Standards Association	视频电子标准协会
VF	video floppy	视频软[磁]盘
VGA	video graphics array	视频图形阵列
VHDL	VHSIC hardware description language	超高速集成电路硬件描述语言
VHDSL	very high bit rate digital subscriber line	甚高速率数字用户专用线
VHLL	very high level language	甚高级语言
VHSIC	very high speed integrated circuit	超高速集成电路
VL Bus	VESA local bus	VL[局部]总线
VLIW	very long instruction word	超长指令字
VLM	virtual Lisp machine	虚拟 Lisp 机
VLR	visitor location register	访问位置寄存器
VLSI	very large scale integration	超大规模集成[电路]
VMD	virtual manufacturing device	虚拟制造设备
VME	virtual machine environment	虚拟机环境
VMS	Virtual Memory Operating System	虚拟存储器操作系统
VODER	voice demonstrator	电子言语合成器
VP	vector processor	向量处理机
VP	verb phrase	动词短语
VP	virtual path	虚通路
VPI	virtual path identifier	虚通路标识符
VPN	virtual private network	虚拟专网
VR	virtual reality	虚拟现实
VRAM	video random access memory	视频随机存取存储器
VRC	vertical redundancy check	垂直冗余检验
VRML	Virtual Reality Modeling Language	虚拟现实建模语言
VRTX	versatile real-time executive	通用实时管理程序
VSAT	very small aperture satellite terminal	甚小孔径卫星终端
VT	virtual terminal	虚拟终端
VTP	virtual terminal protocol	虚[拟]终端协议
VTR	video tape recorder	磁带录像机
VTS	virtual terminal service	虚[拟]终端服务

缩略语	英文全名	中文译名
WAE	wireless application environment	无线应用环境
WAIS	wide area information server	广域信息服务系统
WAM	Warren abstract machine	Warren 抽象机
WAN	wide area network	广域网
WAP	wireless application protocol	无线应用协议
WARC	World Administrative Radio Conference	世界无线电行政会议
WBS	work breakdown structure	工作分解结构
WC	world coordinate system	世界坐标系
W3C	World Wide Web Consortium	万维网联盟
WDM	wavelength division multiplexing	波分多路复用
WDP	Wireless Datagram Protocol	无线数据报协议
WEP	wired equivalent privacy	有线相等的秘密
WG	working group	工作组
WIMP	window-icon-menu-pointing device	WIMP 设备,WIMP(技术)
WLP	wafer level package	圆片级封装
WMF	Windows Metafile format	Windows 元文件格式
WORM	write once read many	一写多读(光盘)
WOSA	Windows open system architecture	Windows 开放系统结构
WP	weakest precondition	最弱前置条件
WPAN	wireless personal area network	无线私人网
WSI	wafer scale integration	圆片规模集成(电路)
WSP	wireless session protocol	无线会话协议
WTLS	wireless transport layer security	无线运输层安全(协议)
WTP	wireless transaction protocol	无线事项处理协议
WWW	world wide web	万维网
WYSIWIS	what you see is what I see	你所见即我所见
XDR	external data representation	外部数据表示
XIP	execute in place	就地执行
XML	extensible markup language	可扩展置标语言
XNOS	experimental network operating system	实验性网络操作系统
XPG	X-Open portability guideline	X - Open 可移植性指南
XPRS	a data extraction, processing and restructuring system	一种数据析取、处理和重构系统
XTI	X-Open transmission interface	X - Open 传输接口
YACC	yet another compiler-compiler	另一种编译程序的编译程序
ZBR	zone bit recording	区位记录
ZCAV	zone constant angular velocity	分区恒角速度
ZM	zero modulation	零调制(码)

附录Ⅱ　计算机及相关学科科技期刊

中 国 期 刊

期刊名称	主　办　单　位	国际刊号 ISSN
计算机学报*	中国计算机学会,中科院计算技术研究所	0254—4164
计算机科学与技术(英文)*	中科院计算技术研究所,中国计算机学会	1000—9000
计算机辅助设计与图形学学报*	中国计算机学会,北京中科期刊出版有限公司	1003—9775
计算机研究与发展*	中国科学院计算技术研究所,中国计算机学会	1000—1239
软件学报*	中国科学院软件研究所,中国计算机学会	1000—9825
计算机工程与设计*	中国航天科工集团 706 所	1000—7024
计算机工程与科学*	国防科技大学计算机学院	1007—130X
计算机科学与探索*	华北计算技术研究所	1673—9418
小型微型计算机系统*	中国科学院沈阳计算技术研究所	1000—1220
计算机科学*	国家科技部西南信息中心	1002—137X
计算机技术与发展*	陕西省计算机学会	1673—629X
计算机工程与应用*	华北计算技术研究所	1002—8331
计算机应用*	中国科学院成都分院、四川省计算机学会	1001—9081
计算机应用研究	四川省计算机研究院	1001—3695
计算机技术	华北计算技术研究所	1002—8846
计算机应用与软件	上海市计算技术研究所,上海计算机软件技术开发中心	1000—386X
工业控制计算机	江苏省计算技术研究所	1001—182X
微电子学与计算机	中国航天科技集团公司九院七七一所	1000—7180
微机发展	陕西省计算机学会	1673—629X
数据通信	信息产业部数据通信技术研究所	1002—5057
微型计算机	科技部西南信息中心	1002—140X
中国科学(F 辑 · 信息科学)	中国科学院、国家自然科学基金委员会	1674—7267
计算机科学前沿(英文)	高等教育出版社、北京航空航天大学	2095—2228
清华大学学报自然科学版(英文)	清华大学	1007—0214
计算可视媒体(英文)	清华大学	2096—0433
大数据挖掘与分析(英文)	清华大学	2096—0654

* 为中国计算机学会会刊。

期刊名称	主办单位	国际刊号 ISSN
计算科学评论	中国大学出版社协会	1673—5153
信息技术与网络安全	华北计算机系统工程研究所	1674—7720
自动化学报	中国自动化学会,中国科学院自动化研究所	0254—4156
图学学报	中国图学学会	2095—302X
计算机产品与流通	天津市电子计算机研究所	1671—1939
计算机集成制造系统	中国兵器工业集团公司第210研究所	1006—5911
计算机教育	清华大学	1672—5913
计算机与数字工程	中国船舶重工集团公司第709研究所	1672—9722
机器人	中国科学院沈阳自动化研究所,中国自动化学会	1002—0446
软件世界	中国电子信息产业发展研究院中国中电报发展有限公司	1005—2348
数值计算与计算机应用	中国科学院数学与系统科学研究院	1000—3266
微计算机应用	中国科学院声学研究所	1003—1944
计算机与信息处理标准化	华北计算技术研究所	1002—8307
中文信息学报	中国中文信息学会,中国科学院软件研究所	1003—0077
微电脑世界	中国计算机世界出版服务公司	1006—8708
计算数学	中国科学院数学与系统科学研究院	0254—7791
软件	中国电子学会,天津电子学会	1003—6970
微小型计算机开发与应用	天津市电子计算机研究所	1001—8786
微处理机	中国电子科技集团公司第四十七研究	1002—2279
计算机工程	华东计算技术研究所,上海市计算机学会	1000—3428
电子计算机外部设备	中国电子科技集团公司第52研究所	1671—7457
计算机时代	浙江省计算技术研究所,浙江省计算机学会	1006—8228
计算机应用文摘	国家科技部西南信息中心	1002—1353
中国计算机用户	中国电子信息产业发展研究院	1003—031X
电脑学习	哈尔滨工业大学等	1002—2422
福建电脑	福建省计算机学会,福建省计算中心	1673—2782
电子与电脑	电子工业出版社	1000—1077
计算机与现代化	江西省计算机学会,江西省计算技术研究所	1006—2475
模式识别与人工智能	中国自动化学会,国家智能计算机研究开发中心等	1003—6059
计算技术与自动化	中国自动化学会,湖南大学	1003—6199
计算机辅助工程	上海海事大学	1006—0871
计算机安全	信息产业部基础产品发展研究中心	1671—0428
计算机网络世界	中国计算机用户协会网络分会	1560—1994
电脑开发与应用	北方自动控制技术研究所	1003—5850
微计算机信息	中国计算机用户协会自动化分会	1008—0570
中国图象图形学报	中国科学院遥感与数字地球研究所,中国图像图形学会等	1006—8961

期刊名称	主 办 单 位	国际刊号 ISSN
信息与电脑	北京方略信息科技有限公司	1003—9767
信息与控制	中国科学院沈阳自动化研究所,中国自动化学会	1002—0411
信息系统工程	天津市信息中心	1001—2362
计算机测量与控制	中国计算机自动测量与控制技术协会	1671—4598
计算机系统应用	中国科学院软件研究所	1003—3254
多媒体世界	国家信息中心	1005—2879
CAD/CAM 与制造业信息化	机械工业信息研究院	1671—8186
电脑爱好者	中国科学院计算技术研究所	1005—0043
计算机仿真	中国航天科工集团公司第 17 研究所	1006—9438
微型电脑应用	上海微型电脑应用学会	1007—757X
电脑与信息技术	中国电子学会,湖南省电子研究所	1005—1228
电脑编程技巧与维护	中国信息产业商会	1006—4052

外国期刊

期刊名称	出版单位或主办单位	国际刊号 ISSN
ACM Computing Surveys	ACM Press	0360—0300
ACM Letters on Programming Languages and Systems	ACM Press	1057—4514
ACM SIGACT News	ACM Press	0163—5700
ACM SIGADA Ada Letters	ACM Press	0736—721X
ACM SIGAPL APL Quote Quad	ACM Press	0163—6006
ACM SIGARCH Computer Architecture News	ACM Press	0163—5964
ACM SIGART Bulletin	ACM Press	0163—5719
ACM SIGCAS Computers and Society	ACM Press	0095—2737
ACM SIGCHI Bulletin	ACM Press	0736—6906
ACM SIGCOMM Computer Communication Review	ACM Press	0146—4833
ACM SIGCPR Computer Personnel	ACM Press	0160—2497
ACM SIGCPR Newsletter	ACM Press	0160—2497
ACM SIGCUE Outlook	ACM Press	0163—5735
ACM SIGFORTH Newsletter	ACM Press	1047—4544
ACM SIGGRAPH Computer Graphics	ACM Press	0097—8930
ACM SIGIR Forum	ACM Press	0163—5840
ACM SIGMALL-PC Notes	ACM Press	0163—5816

期 刊 名 称	出版单位或主办单位	国际刊号 ISSN
ACM SIGMETRICS Performance Evaluation Review	ACM Press	0163—5999
ACM SIGMIS Database	ACM Press	0095—0033
ACM SIGMOD Record	ACM Press	0163—5808
ACM SIGNUM Newsletter	ACM Press	0163—5778
ACM SIGOIS Bulletin	ACM Press	
ACM SIGOPS Operating Systems Review	ACM Press	0163—5980
ACM SIGPLAN Fortran Forum	ACM Press	
ACM SIGPLAN Lisp Pointers	ACM Press	1045—3563
ACM SIGPLAN Notices	ACM Press	0362—1340
ACM SIGSAC Review	ACM Press	
ACM SIGSAM Bulletin	ACM Press	0163—5824
ACM SIGSIM Simulation Digest	ACM Press	0163—6103
ACM SIGSOFT Software Engineering Notes	ACM Press	0163—5948
ACM SIGUCCS Newsletter	ACM Press	0736—6892
ACM Transactions on Computer Systems	ACM Press	0734—2071
ACM Transactions on Computer-Human Interaction	ACM Press	1073—0516
ACM Transactions on Database Systems	ACM Press	0362—5915
ACM Transactions on Graphics	ACM Press	0730—0301
ACM Transactions on Information Systems	ACM Press	1046—8188
ACM Transactions on Mathematical Software	ACM Press	0098—3500
ACM Transactions on Modeling and Computer Simulation	ACM Press	1049—3301
ACM Transactions on Programming Languages and Systems	ACM Press	0164—0925
ACM Transactions on Software Engineering and Methodology	ACM Press	1049—331X
Acta Informatica	Springer-Verlag New York, Inc. Journal Fulfillment Services Dept.	0001—5903
Adaptive Behavior	MIT Press	1059—7123
Advances in Applied Mathematics	Academic Press, Inc.	0196—8858
Advances in Engineering Software	Elsevier Science Ltd.	0965—9978
AI & Society	Springer-Verlag New York, Inc.	0951—5666
AI Expert (Artificial Inteligence)	Miller Freeman Publications Co.	0888—3785
AI Magazine	American Association for Artificial Intelligence	0738—4602

期 刊 名 称	出版单位或主办单位	国际刊号 ISSN
Algorithmica; An International Journal in Computer Science	Springer-Verlag New York, Inc.	0178—4617
Analog Integrated Circuits and Signal Processing	Kluwer Academic Publishers	0925—1030
Applied Artificial Intelligence	Taylor & Francis/Hemisphere	0883—9514
Applied Mathematics and Computation	Elsevier Science, Inc.	0096—3003
Applied Networks Report	International News Group	0899—5680
Applied Numerical Mathematics	Elsevier Science Publishers B. V. The Netherlands	0168—9274
Artificial Intelligence	Elsevier Science Publishers B. V. The Netherlands	0004—3702
Artificial Intelligence and Law	Kluwer Academic Publishers	0924—8463
Artificial Intelligence Review	Kluwer Academic Publishers	0269—2821
Artificial Life	MIT Press	1064—5462
ASLIB Proceedings	The Association for Information Management(UK)	0001—253X
Australian Computer Journal	Australian Computer Society, Inc.	0004—8917
Automated Software Engineering	Kluwer Academic Publishers	
The Bulletin of Symbolic Logic	Association for Symbolic Logic, Inc.	1079—8986
Byte	McGraw-Hill Publication, Inc.	0360—5280
C++ Report	SIGS Publications, Inc.	1040—6042
C/C++ Users Journal	R & D Publications, Inc.	1075—2838
Canadian Information Processing/ Informatique Canadienne	Canadian Information Processing Society	1182—3097
CASE Trends(Computer-Aided Software Engineering)	Software Productivity Group, Inc.	1046—5944
CD-ROM World	Mecklermedia Corp.	1066—274X
Cipher	IEEE Press	
Circuits, Systems, and Signal Processing	Birkhäuser Boston, Inc.	0278—081X
Client/Server Computing	Sentry Publishing Company, Inc.	1059—3470
Client/Server Today	Cahners Publishing Company	1078—8565
Collegiate Microcomputer	Rose-Hulman Institute of Technology	0731—4213
Communications of the ACM	ACM Press	0001—0782
Computational Complexity	Birkhäuser Boston, Inc.	1016—3328
Computational Economics	Kluwer Academic Publishers	0927—7099
Computational Geometry: Theory and Applications	Elsevier Science Publishers B. V. The Netherlands	0925—7721
Computational Linguistics	MIT Press	0891—2017
Computational Mathematics and Mathematical Physics	Elsevier Science, Inc.	0965—5425

期 刊 名 称	出版单位或主办单位	国际刊号 ISSN
Computational Optimization and Applications	Kluwer Academic Publishers	0926—6003
Computational Statistics & Data Analysis	Elsevier Science Publishers B. V. The Netherlands	0167—9473
Computer	IEEE Computer Society Press	0018—9162
Computer Aided Design	Butterworth-Heinemann Turpin Transactions Ltd. (UK)	0031—3202
Computer-Aided Engineering	Penton Business Publishing	0733—3536
Computer Aided Geometric Design	Elsevier Science Publishers B. V. The Netherlands	0167—8396
Computer Design	Penn Weli Publishing Co.	0010—4566
Computer Fraud and Security	Elsevier Advanced Technology(UK)	1361—3723
Computer Graphics Forum	Blackwell Publishers	0167—7055
The Computer Journal	Oxford University Press(UK)	0010—4620
Computer Languages	Pergamon Press, Inc. Imprint of Elsevier Science Inc.	0096—0551
Computer Language	Computer Language	0747—2839
Computer literature Index	Applied Computer Research, Inc.	0270—4846
Computer Networks and ISDN Systems	Elsevier Science Publishers B. V. The Netherlands	0169—7552
Computer Processing of Chinese and Oriental Languages	USA	0715—9048
Computer Optimization and Applications	Kluwer Academic Publishers	0926—6003
Computer Pictures	Montage Publishing, Inc.	0883—5683
Computer Report	日本经营科学研究所	0385—6658
Computer Science and Informatics	Computer Society of India	0254—7813
Computer Security Journal	Computer Security Institute	0277—0865
Computer Speech and Language	Harcourt Brace Jovanovich Ltd. (UK)	0885—2308
Computer Standards and Interfaces	Elsevier Science Publishers B. V. The Netherlands	0920—5489
Computer Supported Cooperative Work	Kluwer Academic Publishers	0925—9724
Computer Systems Science and Engineering	CRL Publishing Ltd. (UK)	0267—6192
Computer Vision and Image Understanding	Academic Press, Inc.	1077—3142
Computers and Artificial Intelligence	Inst. of Computer Systems	0232—0274
Computers and Biomedical Research	Academic Press, Inc.	0010—4809

期 刊 名 称	出版单位或主办单位	国际刊号 ISSN
Computers and Education	Pergamon Press, Inc. Imprint of Elsevier Science Inc.	0360—1315
Computers and Electrical Engineering	Pergamon Press, Inc. Imprint of Elsevier Science Inc.	0045—7906
Computers and Fluids	Pergamon Press, Inc. Imprint of Elsevier Science Inc.	0045—7930
Computers and Geosciences	Pergamon Press, Inc. Imprint of Elsevier Science Inc.	0098—3004
Computers and Graphics	Elsevier Science Publishers	0097—8493
Computers and Industrial Engineering	Pergamon Press, Inc. Imprint of Elsevier Science Inc.	0360—8352
Computers and Operations Research	Pergamon Press, Inc. Imprint of Elsevier Science Inc.	0305—0548
Computers and Security	Elsevier Advanced Technology Publications(UK)	0167—4048
Computers in Human Services	Haworth Press, Inc.	0740—445X
Computers in Industry	Elsevier Science Publishers B. V. The Netherlands	0166—3615
Computers in Physics	American Institute of Physics, Inc.	0894—1866
Computing	Springer-Verlag New York, Inc.	0010—485X
Computing Reviews	ACM Press	0010—4884
Computing Systems	MIT Press	0895—6340
Data & Knowledge Engineering	Elsevier Science Publishers B. V. The Netherlands	0169—023X
Database	Online, Inc.	0162—4105
Database and Network Journal	A. P. Publications Ltd.	0308—3314
Data Base Product Report	Management Information Corp.	0740—6800
Database Programming & Design	Miller Freeman Publications	0895—4518
Data Communications International	McGraw Hill Publication, Inc.	
Datamation	Cahners Publishing Company	0011—6963
DBMS	Miller Freeman Publications	1041—5173
Decision Support Systems	Elsevier Science, Inc.	0167—9236
Discrete Applied Mathematics	Elsevier Science Publishers B. V. The Netherlands	0166—218X
Discrete Mathematics	Elsevier Science Publishers B. V. The Netherlands	0012—365X
Distributed and Parallel Databases	Kluwer Academic Publishers	0926—8782
Distributed Computing	Springer-Verlag(Germany)	0178—2770
Distributed Systems Engineering Journal	IOP Publishing Ltd. ,(UK)	0967—1846
Electronic Publishing—Origination, Dissemination, and Design	John Wiley and Sons Ltd.	0894—3982
Expert Systems	Learned Information Ltd.	

期 刊 名 称	出版单位或主办单位	国际刊号 ISSN
Finite Elements in Analysis and Design	Elsevier Science Publishers B. V. The Netherlands	0168—874X
Formal Methods in System Design	Kluwer Academic Publishers	0925—9856
Future Generation Computer Systems	Elsevier Science Publishers B. V. The Netherlands	0167—739X
Fuzzy Sets and Systems	Elsevier Science Publishers B. V. The Netherlands	0165—0114
Graphical Models and Image Processing	Academic Press, Inc.	1077—3169
Home Computer Magazine	Emerald Valley Publishing Co.	0747—055X
IBM Journal of Research and Development	IBM Corp.	0018—8646
IBM Systems Journal	IBM Corp.	0018—8670
IEE Proceedings—Computers and Digital Techniques	IEE	1350—2387
IEE Proceedings—Vision	Image and Signal Processing IEE	
IEEE/ACM Transactions on Networking	ACM Press	1063—6692
IEEE Annals of the History of Computing	IEEE Computer Society Press	1058—6180
IEEE Computational Science & Engineering	IEEE Computer Society Press	1070—9924
IEEE Computer Graphics and Applications	IEEE Computer Society Press	0272—1716
IEEE Design & Test	IEEE Computer Society Press	0740—7475
IEEE Micro	IEEE Computer Society Press	0272—1732
IEEE MultiMedia	IEEE Computer Society Press	1070—986X
IEEE Network	IEEE Press	0890—8044
IEEE Parallel & Distributed Technology: Systems & Technology	IEEE Computer Society Press	1063—6552
IEEE Software	IEEE Computer Society Press	0740—7459
IEEE Spectrum	IEEE Press	0018—9235
IEEE Transactions on Computers	IEEE Press	0018—9340
IEEE Transactions on Parallel & Distributed Systems	IEEE Press	1045—9219
IEEE Transactions on Pattern Analysis and Machine Intelligence	IEEE Press	0162—8828
IEEE Transactions on Software Engineering	IEEE Press	0098—5589

期 刊 名 称	出版单位或主办单位	国际刊号 ISSN
IEEE Transactions on Very Large Scale Integration (VLSI) Systems	IEEE Press	1063—8210
IFIP Transaction, A: Computer Science and Technology	Elsevier Science Publishers	0926—5473
Image Understanding	Academic Press	1049—9660
IMPACT of Computing in Science and Engineering	Academic Press, Inc.	0899—8248
Information and Computation	Academic Press, Inc.	0890—5401
Information and Management	Elsevier Science, Inc.	0378—7206
Information Processing and Management: An International Journal	Pergamon Press, Inc. Imprint of Elsevier Science Inc.	0306—4573
Information Processing Letters	Elsevier Science Publishers B. V. The Netherlands	0020—0190
Information Resources Management Journal	Idea Group Publishing	1040—1628
Information Sciences—Applications: An International Journal	Elsevier Science, Inc.	1069—0115
Information Sciences—Informatics and Computer Science: An International Journal	Elsevier Science, Inc.	0020—0255
Information Sciences—Intelligent Systems: An International Journal	Elsevier Science, Inc.	0020—0255
Information Systems	Pergamon Press, Inc.	0306—4379
Integration, the VLSI Journal	Elsevier Science Publishers B. V. The Netherlands	0167—9260
Intelligent Software Strategies	Cutter Information Corp.	0887—221X
Interacting with Computers	Butterworth-Heinemann	0953—5438
International Journal in Computer Simulation	Ablex Publishing Corp.	1055—8470
International Journal of Applied Software Technology	International Academic Publishing Co. (Canada)	1198—5577
International Journal of Computer Mathematics	Gordon and Breach Science Publishers	0027—7160
International Journal of Computer Systems	CRL	
International Journal of Computer Systems Science and Engineering	CRL	
International Journal of Computer Vision	Kluwer Academic Publishers	0920—5691

期 刊 名 称	出版单位或主办单位	国际刊号 ISSN
International Journal of Expert Systems	JAI Press, Inc.	0894—9077
International Journal of Foundations of Computer Science	World Scientific Publishing Co. (Singapore)	0129—0541
International Journal of Game Theory	Springer-Verlag New York, Inc. Journal Fulfillment Services Dept.	0020—7276
International Journal of High-Speed Computing	World Scientific Publishing Co. (Singapore)	0129—0533
International Journal of Human—Computer Interaction	Ablex Publishing Corp.	1044—7318
International Journal of Human—Computer Studies	Academic Press, Inc. (UK)	1071—5819
International Journal of Intelligent Systems	John Wiley and Sons, Inc.	0884—8173
International Journal of Man—Machine Studies	Academic Press, Inc. (UK)	0020—7373
International Journal of Mini and Microcomputers	ISMM	0702—0481
International Journal of Parallel Programming	Plenum Press Imprint of Plenum Publishing Corp.	0885—7458
International Journal of Pattern Recognition and Artificial Intelligence	World Scientific Publishing Co. (Singapore)	0218—0014
International Journal of Robotics Research	MIT Press	0278—3649
International Journal of Software Engineering and Knowledge Engineering	World Scientific Publishing Co. (Singapore)	0218—1940
International Journal of Supercomputer Applications and High Performance Engineering	MIT Press	0890—2720
International Journal of Vitrual Reality	N. Cascade Avenue, Colorado Springs	1081—1451
Internet Research	MCB University Press	1066—2243
Journal of Algorithms	Academic Press, Inc.	0196—6774
Journal of Artificial Intelligence in Education	Association for the Advancement of Computing in Education	1043—1020
Journal of Artificial Neural Networks	Ablex Publishing Corp.	1073—5828
Journal of Automated Reasoning	Kluwer Academic Publishers	0168—7433
Journal of Circuits, Systems and Computers	World Scientific Publishing	

期 刊 名 称	出版单位或主办单位	国际刊号 ISSN
Journal of Computational & Applied Mathematics	Elsevier Science Publishers B. V. The Netherlands	0377—0427
Journal of Computer and Software Engineering	Ablex Publishing Corp.	1069—5451
Journal of Computer and System Sciences	Academic Press, Inc.	0022—0000
Journal of Database Management	Idea Group Publishing	1063—8016
Journal of Educational Multimedia and Hypermedia	Association for the Advancement of Computing in Education	1055—8896
Journal of Electronic Testing Theory and Applications	Kluwer Academic Publishers	0923—8174
Journal of End User Computing	Idea Group Publishing	1063—2239
Journal of Experimental & Theoretical Artificial Intelligence	Taylor & Francis, Inc.	0952—813X
Journal of High Speed Networks	IOS Press	0926—6801
Journal of Graph Theory	John Wiley & Sons, Inc.	0364—9024
Journal of Information Processing	日本情报处理学会	0387—6101
Journal of Information Science	Bowker-Saur Limited(UK)	0165—5515
Journal of Intelligent Information Systems	Kluwer Academic Publishers	0925—9902
Journal of KISS(A)(Computer Systems and Theory)	Korea Information Science Society	
Journal of KISS(B)(Software and Applications)	Korea Information Science Society	
Journal of KISS(C)(Computing Practices)	Korea Information Science Society	
Journal of Logic Programming	Elsevier Science, Inc.	0743—1066
Journal of Management Information Systems	M. E. Sharpe, Inc.	0742—1222
Journal of Mathematical Imaging and Vision	Kluwer Academic Publishers	0924—9907
Journal of Microcomputer Applications	Academic Press, Inc. (UK)	0745—7138
Journal of Network and Systems Management	Plenum Press Imprint of Plenum Publishing Corp.	1064—7570
Journal of New Generation Computer Systems	STBS Ltd.	0863—0445
Journal of Parallel and Distributed Computing	Academic Press, Inc.	0743—7315
Journal of Scientific Computing	Plenum Press Imprint of Plenum Publishing Corp.	0885—7474
The Journal of Supercomputing	Kluwer Academic Publishers	0920—8542

期 刊 名 称	出版单位或主办单位	国际刊号 ISSN
Journal of Symbolic Computation	Academic Press, Inc. (UK)	0747—7171
Journal of Symbolic Logic	Association for Symbolic Logic, Inc.	0022—4812
Journal of Systems and Software	Elsevier Science, Inc.	0164—1212
Journal of the ACM	ACM Press	0004—5411
Journal of the American Society for Information Science (JASIS)	John Wiley & Sons, Inc.	0002—8231
Journal of the Computer Society of India	Computer Society of India	0045—7892
Journal of VLSI Signal Processing	Kluwer Academic Publishers	0922—5773
Knowledge Acquisition	Academic Press, Inc. (UK)	1042—8143
Knowledge-based Systems	Butter Worth-Heinemann Turpin Transactions Ltd.	0950—7051
Languages of Design	Elsevier Science Publishers B. V.	0927—3034
Lisp and Symbolic Computation	Kluwer Academic Publishers	0892—4635
Local Area Network (LAN)	Information Gatekeepers, Inc.	1051—1962
Machine Learning	Kluwer Academic Publishers	0885—6125
Machine Vision and Applications	Springer-Verlag New York, Inc.	0932—8092
Management Science	The Institute for Operations Research and Management Sciences	0025—1909
Mathematical Programming	Elsevier Science Publishers B. V. The Netherlands	0025—5610
Mathematics of Computation	American Mathematical Society	0025—5718
Methods of Logic in Computer Science	Ablex Publishing Corp.	1075—0924
Microelectronic Engineering	Elsevier Science Publishers B. V. The Netherlands	0167—9317
Microprocessing and Microprogramming	Elsevier Science Publishers B. V. The Netherlands	0165—6074
Microprocessors & Microsystems	Butterworth-Heinemann	0141—9331
Micro-Systems	Society Parisienne d'Edition (France)	
MIS Quarterly	Society for Information Management and The Management Information Systems Research Center	0276—7783
Multidimensional Systems and Signal Processing	Kluwer Academic Publishers	0923—6082
Multimedia Systems	Springer-Verlag New York, Inc.	0942—4962
Multimedia Systems	Information Europe Ltd.	0960—7749
Multimedia Tools and Applications	Kluwer Academic Publishers	1380—7501
Network Computing	CMP Publications, Inc.	1046—4468
Networks	Wiley-Interscience	0028—3045
Neural Computation	MIT Press	0899—7667
Neural Networks	Pergamon Press, Inc.	0893—6080

期刊名称	出版单位或主办单位	国际刊号 ISSN
Neural, Parallel & Scientific Computations	Dynamic Publisher, Inc.	1061—5369
New Generation Computing	Springer-Verlag New York, Inc.	0288—3635
New Review of Applied Expert Systems	Taylor Graham Publishing(UK)	1361—0244
New Review of Hypermedia and Multimedia	Applications and Research(UK)	1361—4568
On The Internet	Rickard Associates	1081—3969
OS 2 Professional	I. F. Computer Media, Inc.	1069—6814
Packaged Software Report; CASE Product Report	Management Information Corp.	0747—9573
Parallel Algorithms and Applications	STBS Ltd. Order Department	1063—7192
Parallel Computing	Elsevier Science Publishers B. V. The Netherlands	0167—8191
Pattern Recognition	Elsevier Science Ltd.	0031—3203
Pattern Recognition Letters	Elsevier Science Publishers B. V. The Netherlands	0167—8655
PC Magazine	Ziff. Davis Publishing Co.	0888—8507
Performance Evaluation	Elsevier Science Publishers B. V. The Netherlands	0166—5316
Presence: Teleoperators and Virtual Environments	MIT Press	1054—7460
Real-Time Systems	Kluwer Academic Publishers	0922—6443
Resource Sharing and Information Networks	Haworth Press, Inc.	0737—7797
Science of Computer Programming	Elsevier Science Publishers	0167—6423
Scientific Programming	John Wiley & Sons, Inc.	1058—9244
SIAM Journal on Applied Mathematics	Society for Industrial and Applied Mathematics	0036—1399
SIAM Journal on Computing	Society for Industrial and Applied Mathematics	0097—5397
SIAM Journal on Control and Optimization	Society for Industrial and Applied Mathematics	0353—0129
SIAM Journal on Discrete Mathematics	Society for Industrial and Applied Mathematics	0895—4801
SIAM Journal on Mathematical Analysis	Society for Industrial and Applied Mathematics	0036—1410
SIAM Journal on Matrix Analysis and Applications	Society for Industrial and Applied Mathematics	0895—4798
SIAM Journal on Numerical Analysis	Society for Industrial and Applied Mathematics	0036—1429

期 刊 名 称	出版单位或主办单位	国际刊号 ISSN
SIAM Journal on Scientific Computing	Society for Industrial and Applied Mathematics	1064—8275
Signal Processing	Elsevier Science Publishers B. V. The Netherlands	0165—1684
Simulation	Society for Computer Simulation	0037—5497
Simulation and Gaming	Sage Publications, Inc.	1046—8781
Simulation Practice and Theory	Elsevier Science Publishers B. V. The Netherlands	0928—4869
Sixth Generation Systems	Gallifrey Publishers	
Social Science Computer Review	Duke University Press	0894—4393
Software Concepts and Tools	Springer-Verlag	
Software Development	Miller Freeman Publications	1070—8588
Software Engineering Journal	Institution of Electrical Engineers(UK)	0268—6961
Software—Industry Report	Computer Age and EDP News Services	
Software Law Journal	Center for Computer/Law	
Software Magazine	Sentry Publishing Company, Inc.	0897—8085
Software Management News	Software Maintenance News, Inc.	0741—4501
Software—Practice & Experience	John Wiley & Sons, Inc.	0038—0644
Software Quality Journal	Chapman and Hall (UK)	
Software Testing, Verification and Reliability	John Wiley and Sons Ltd. (UK)	
Speech Communication	Elsevier Science Publishers B. V. The Netherlands	0167—6393
Standard View	ACM Press	1067—9936
System and Computers in Japan	John Wiley and Sons, Inc.	0882—1666
Systems Analysis Modelling Simulation	Gordon and Breach Science Publishers, Inc.	0232—9298
Systems Integration Business	Cahners Publishing Co.	0364—9342
Theoretical Computer Science	Elsevier Science Publishers B. V. The Netherlands	0304—3975
Theory and Practice of Object Systems	John Wiley & Sons, Inc.	1074—3227
Topology and Its Applications	Elsevier Science Publishers B. V. The Netherlands	0166—8641
Transaction of Information Processing Society of Japan	Information Processing Society of Japan	0387—5806
Transactions of the Society for Computer Simulation	Society for Computer Simulation	0740—6797
The Visual Computer	Springer-Verlag New York, Inc. Journal Fulfillment Services Dept.	0178—2789
Unix Review	Miller Freeman Publications Co.	0742—3136
Visual Computer	Springer-Verlag(Germany)	0178—2789

期 刊 名 称	出版单位或主办单位	国际刊号 ISSN
Visualization and Computer Animation	Wiley Publishers	1049—8907
VLSI Design	Gordon and Breach	0279—2837
Windows User	Wandsworth Publishing, Inc.	1065—3481
Windows/DOS Developer's Journal	R & D Publications, Inc.	1059—2407
Wireless Networks	ACM Press	1022—0038
Wireless Personal Communications: An International Journal	Kluwer Academic Publishers	0929—6212
Wirtschafts Informatik	Friedr. Vieweg & Sohn Verlags-gesellschaft mbft (Germany)	0937—6429
コンピューター　ソフトウェア (*Computer Software*)	岩波书店情报处理,日本情报处理学会	0447—8053
Автоматика и вычислительная техника	Международная Книга	0132—4160
Вычислительные машины и системы, выпуск "РЖ"	Международная Книга, Москва	0234—9663
Вычислительные науки, отдельный выпуск "РЖ"	Международная Книга, Москва	0235—1501
Вычислительная техника (ЭИ)	Международная Книга, Москва	0132—1730
Программирование	Международная Книга, Москва	0132—3474
Микроэлектроника	Москва, Российская Академика Наук	0544—1269

主要参考资料

1. Source Publishions Lists. Science Citation Index (SCI). 3D. Institute of Science Information, Inc., 1996
2. INSPEC List of Journals. Computer and Control Abstracts. Science Abstracts Series C, December, 1995, and № 1～8, 1996, INSPEC
3. Computing Reviews. November, 1995, 599～608
4. Computer Abstracts. 1995, 39: 930～931
5. 外国报刊目录. 万国学术出版社, 1993, 1995
6. 报刊简明目录. 北京报刊发行局, 1997
7. 北京图书馆期刊目录
8. 中国科学院文献情报中心期刊目录

（张　伟）

附录Ⅲ　计算机及相关学科学术团体

1. 中国计算机学会,China Computer Federation (CCF)
   成立于 1962 年.
   地址：中国,北京市海淀区中关村科学院南路 6 号,100190,北京 2704 信箱.
   China, Beijing 100190, P. O. Box 2704.
   电话：86-010-62562503　　传真：86-010-62527485
   E-mail：ccf @ ccf. org. cn

2. 中国自动化学会,Chinese Association of Automation (CAA)
   成立于 1961 年 11 月.
   地址：中国,北京市海淀区中关村东路 95 号,100190,北京 2728 信箱.
   China, Beijing 100190, P. O. Box 2728.
   电话：86-010-82544542　　传真：86-010-62522248
   E-mail：cca@ ia. ac. cn

3. 中国通信学会,China Institute of Communications (CIC)
   成立于 1980 年 12 月.
   地址：中国,北京西长安街 13 号,100804.
   China, Beijing 100804.
   电话：86-010-68209079　　传真：86-010-68209074
   E-mail：zhufeng@ china-cic. cn

4. 中国中文信息学会,Chinese Information Processing Society of China (CIPS)
   成立于 1981 年 6 月.
   地址：中国,北京市海淀区中关村南四街 4 号,100190, 北京 8718 信箱.
   China, Beijing 100080, P. O. Box 8718.
   电话：86-010-62562916　　传真：86-010-62661046
   E-mail：cips@ iscas. ac. cn

5. 中国电子学会,Chinese Institute of Electronics (CIE)
   成立于 1962 年.
   地址：中国,北京市海淀区玉渊潭南路普惠南里 13 号楼,100036,北京 165 信箱.
   China, Beijing 100036, P. O. Box 165.
   电话：86-010-68283461　　传真：86-010-68283458

6. 中国图象图形学会,China Society of Image and Graphics
   成立于 1990 年.
   地址：中国,北京市海淀区中关村东路 95 号东楼 307,100190.

China, Beijing 100190.
电话(传真): 86-010-82544676

7. 香港电脑学会, Hong Kong Computer Society (HKCS)
成立于 1970 年.
地址: 中国, 香港九龙塘达之路 78 号 5 层.
5F, 78 Tat Chee Avenue, Kowloon Tong, Hong Kong.
电话: 852-2834 2228　传真: 852-2834 3003
E-mail: hkcs @ hkcs. org. hk
http: //www. nkcs. org. hk

8. Taipei Computer Association
成立于 1974 年.
Add: Taipei, 3rd floor, 2 PA teh Road, section 3. China.
Tel: 2-7764249　Fax: 2-7764410
http: //www. tca. org. tw

9. American Association for Artifical Intelligence (AAAI)
Add: 445 Burgess Drive, Menlo Park, California 94025. U. S. A.
Tel: 1-(415)328-3123　Fax: 1-(415)321-4457
http: //www. aaai. org

10. American Federation of Information Processing Societies (AFIPS)
成立于 1961 年.
Add: 1899 Preston White Drive, Reston. VA 22091. U. S. A.
Tel: 1-(703) 620-8900

11. American Sociaty for Information Science (ASIS)
成立于 1937 年. 1968 年改为现名.
Add: N. W. Suite 404, Washington DC 20036. U. S. A.
Tel: 1-(202)462-1000
http: //www. asis. org

12. Associacao Portugesa de Informatica (AIP)
Add: Portugal, Av. Almirante Reis 127, 1°. Esq. P-1100 LISBON.
Tel: 351-(1) 535587　Fax: 351-(1) 570410
Telex: 64653apint p.
http: //www. api. pt

13. Association for Computational Linguistics (ACL)
Add: PO Box 6090 Somerset, NJ 08875, U. S. A.
http: //www. cs. columbia. edu / ~ acl /home. html

14. Association for Computing Machinery (ACM)

成立于 1947 年 9 月 15 日.

Add：1515 Broadway, New York，NY10036. U. S. A.

Tel：1-(212) 6260500　　Fax：1-(212) 9441318

E-mail：acmhelp @ acm. org

http：//www. acm. org

15. Association for Systems Manegement (ASM)

成立于 1947 年.

Add：24587 Begley Road ,Cleveland,ON 44138. U. S. A.

Tel：1-(216) 243-6900

http：//aww. asm. org

16. Association Francaise pour la Cybernetique Economique et Technique(AFCET)

Add：156 boulevard Pereire，F-75017 Paris. France.

Tel：33-(1) 47662419　　Fax：33-(1) 42679312

Telex：283155 F Code 235 E.

17. Association of Data Processing Service Organizations (ADAPSO)

Add：1300 North 17th Street，Arlington,VA 22209. U. S. A.

Tel：1-(703) 522-5055

http：//www. cni. org /docs /infopols /adapso. html ＊

18. Associazione Italiana per L'Informatica ed il Calcolo Automatico(A. I. C. A.)

Add：Piazza R. morandi，2，I-20121 MILAN. Italy.

Tel：39-(2) 784970　　Fax：39-(2) 76014082

19. Australian Computer Society (ACS)

Add：P. O. Box 319，DARLINGHURST,N. S. W. 200. Australian.

Tel：61-(2) 2115855　　Fax：61-(2) 2811208

http：//www. acs. org. au

20. Austrian Computer Society

Add：Wollzeile 1-3,A-1010 VIENNA. Austria.

Tel：43-(1) 5120235　　Fax：43-(1) 5137735

21. Bangladesh Computer Society

成立于 1979 年.

Add：c /o Computer Centre，Bangladesh University of Engineering and Technology,Dhaka1000，Bangladesh.

Tel：880-(2)503744　　Fax：880-(2)863026

http：//www. tuns. ca / ~ abidmr /buet. html

22. British Computer Society (BCS)

成立于 1957 年.

Add：P. O. Bpx 1454，Station Road，Swindon，SNI 1TG，United Kingdom.
Tel：44-(793) 480269 Fax：44-(793) 480270
http：//www. bcs. org. uk

23. Canadian Information Processing Society (CIPS)
Add：430 King St. West，Suite 205，TORONTO，Ontario，Canada M5V 1L5. Canada.
Tel：1-(416) 5934040 Fax：1-(416) 5935184
http：//www. cips. ca

24. Centre Nacional d' Informatica d' Andorra (CNIA)
Add：Avgda. Santa Coloma，91 ANDORRA LA VELLA. Andorra.
Tel：33-(628) 22400 Fax：33-(628) 28218
http：//www. andorra. ad

25. Centre National de L' Informatique
Add：17 rue Belhassan Ben Chaabane，EL Omrane，1005 TUNIS. Tunisia.
Tel：216-(1) 782996 Fax：216-(1)781862
Telex：13904 ceninf tn.
E-mail：kamoun @ tuniscni. uucp

26. Chinese & Oriental language Information Processing Society(COLIPS)
Add：Department of Information System and Computer Science，National University of Singapore. Kent Ridge，Singapore.
Fax：65-(779) 4580
E-mail：luakt @ iscs. nus：sg
http：//www. iscs. nus. sg / ~ colips /commcolips

27. Computer Association of Nigeria (CAN)
Add：P. O. Box 4800，5 Akinhanmi Street，SURULERE，LAGOS. Nigeria.

28. Computer Association of Thailand
成立于 1972 年.
Add：2nd Floor，Chulalongkorn University Alumni Association Building，Phayathai Road，Bangkok 10330，Thailand.
Tel：66-(2)2153546 Fax：66-(2)2153962
http：//www. nectec. or. th

29. Computer Society of India (CSI)
成立于 1964 年.
Add：. c /o Institutionof Ebgineers Building，K. Khadye Marg，15 haji Ali Park，BOMBAY 4000 034. India.
Tel：91-(22) 4943422 or 4949433

30. Computer Society of Pakistan
成立于 1973 年.

Add: Office 5, 3rd Floor, Sasi Arcade, Main Clifton Road, Karachi, Pakistan.
Tel: 92-(21)5701030

31. Computer Society of South Africa
Add: P. O. Box 1714, HALFWAY HOUSE 1685. Rep. of South Africa. South Africa.
Tel: 27-(11) 3151319　　Fax: 27-(11) 3152276

32. Computer Society of Zimbabwe
Add: P. O. Box 8385, Causeway, HARARE. Zimbabwe.
Tel: 263-(4) 706725　　Fax: 263-(4) 725657
Telex: 26433 cmihre zw.

33. Czechoslovak Committee for IFIP
Add: Czechoslovakia. c /o Dr. J. Dolezal (Nat. Corres.), Czech. Academy of Sciences, Pod vodarenskou vezi 4, CS -18208 PRAGUE.
Tel: 42-(2) 8152062　　Fax: 42-(2) 847452
Telex: 122018 atom c.
E-mail: dolezal @ cspgasll

34. Danish Federation for Information Processing (DANFIP)
Add: Kronprinsensgade 14, DK-1114 COPENHAGEN K. Denmark.
Tel: 45-33-111560　　Fax: 45-33-931580

35. European Association for Computer Graphics
Add: EUROGRAPHICS Association, P. O. Box 16, CH-1288 Aire-Ia-Ville, Switzerland.
Fax: 41-(22) 7570318
E-mail: secretart @ eg. org
http: //www. et. org

36. Egyptian Computer Society
Add: P. O. Box 9009, NasrCity, CAIRO, Tth A. R. E. Egypt.
Tel: 20-(2) 601484

37. European Association for Microprocss. and Microprogramming(EUROMICRO)
Add: P. O. Box 2346, NL-7301 EA APELDOORN, Netherlands.
Tel: 31-(55) 557372　　Fax: 31-(55) 557393
E-mail: eruomicro @ standby. nl
http: //europub. ict. tuwien. ac. at

38. FAIB -FBVI
Add: Square de Biarritz 3, Bte 5, B -1050 BRUSSELS. Belgium.
Tel: 32-(2) 2371045　　Fax: 32-(2) 2306785
Telex: 22075.
http: //www. bfia. be

39. Federation Espanola de Sociedades de Informatica (FESI)

Add: Hortaleza 104 -2°. izqda. E-28004 MADRID. Spain.

Tel: 34-(1) 5192565　Fax: 34-(1) 5440763

E-mail: fesi@ dip. upm. es

http: //www. upm. es *

40. Federation on Computing in the United States (FOCUS)

1991 年由 ACM 和 IEEE CS 联合组成,作为 IFIP 的成员.

Add: 1730 Massachusetts Aenue, N. W. WASHINGTON, D. C. 20036-1903. U. S. A.

Tel: 1-(202) 3710101　Fax: 1-(202) 7289614

41. Finnish Information Processing Association

Add: P. O. Box 68, SF-02601 ESPOO. Finland.

Tel: 358-(90) 5121255　Fax: 358-(90) 5121276

http: //www. ttlry. fi /2english. htm

42. Gesellschaft fur Informatik (GI)

成立于 1962 年.

Add: Godesberger Allee 99, D -5300 BONN 2. Germany.

Tel: 49-(228) 376751　Fax: 49-(228) 378178

http: //www. gi-ev. de

43. Greek Computer Society

Add: 2 Mavromichali Street, 106 79 ATHENS. Greece.

Tel: 30-(1) 3645274　Fax: 30-(1) 3645154

http: //www. otenet. gr /epy /index-en. htm

44. ICIMAF

Add: . c /o Mr B. J. Ferro, Academia de Ciencias de Cuba, Calle 15 № 551, Vedado, C. Habana 10400. Rep. de Cuba.

E-mail: icimaf @ ceniai. cu

Telex: 512230 icimaf cu

45. IEEE Computer Society (IEEE CS)

成立于 1951 年.

Add: 1730 Massachusetts Ave. NW, Washington, DC20036-1903. U. S. A.

Tel: 1-(202)371-0101　Fax: 1-(202) 728-9614

E-mail: hg. ofc @ Compmail. com

http: //www. computer. org

46. Indonesian Computer Society

成立于 1974 年.

Add: P. O. Box 2454, Jakarta 10002, Indonesian.

Tel: 62-(21) 5201010　Fax: 62-(21) 5200077

Telex：62779.
http：//indonesian-society. com

47. Information Processing Association of Israel (IPA)
Add：Kfar-Maccabiah, RAMAT-GAN 52109. Israel.
Tel：97-(23) 715770 or 72　　Fax：97-(23) 5744374

48. Information Processing Society of Japan (IPSJ),情报处理学会
Add：108 东京都港区芝浦 3-16-20，芝浦前川 Building,7 阶. 日本国.
Tel：81-(03)5484-3535　　Fax：81-(03)5484-3534
E-mail：edit j @ ipsj. or. jp
http：//www. ipsj. or. jp

49. Institute of Electrical Engineers (IEE)
Add：Savoy Placa, London WC2R OBL UK. United Kingdom.
Tel：44-0171-240 1871　　Fax：44-0171-240 7735
Telex：261176 IEEDN G.
http：//www. iee. org

50. Institute of Electrical and Electronic Engeneers(IEEE)
Add：345 East 47th, New York NY 10017-2394 U. S. A.
http：//www. ieee. org

51. International Association for Mathematics and Computer in Simulation (IMACS)
Add：c /o Dept. of Computer Science, Rutgers Univ. New Brunswick, NJ 08903. U. S. A.
Tel：1-(201)932 3998
E-mail：vichneve @ cs. rutgers. edu
http：//www. cs. rutgers. edu

52. International Association for Pattern Recognition (IAPR)
Add：66 Weston Park, Thames Ditton, Surrey KT7 OHL UK. United Kingdom.
Tel /Fax：44-181 (398) 2766
E-mail：100042. 511 @ compuserve. com
http：//peipa. essex. ac. uk /iapr

53. International Association for Statist. Computing
Add：c /s ISI, 428 Prinses Beatrixlaan, NL-2270 AZ VOORBURG. Netherlands.
Tel：31-(70) 33757737　　Fax：31-(70) 3860025
E-mail：lieves @ cs. vu. nl
http：//fisher. stat. unipg. it /isac

54. International Federation for Information Processing (IFIP)
成立于 1960 年 1 月,它是多国相关学会的联合学术团体,到 1992 年 1 月已有 42 个国家和地区的学术团体成为会员,每个国家只吸收一个会员单位.

Add: Hofstrasse 3, A-2361 Laxenburg, Austria.
Tel: 43-2236-73616 Fax: 43-2236-73616
E-mail: ifip @ ifip. or. at
http: //www. ifip. or. at

55. International Federation of Automatic Control (IFAC)
Add: Schlossplatz 12, A-2361 lazenburg, Austria.
Fax: 43-2236-72859
E-mail: ifaca @ serv. univie ac. at
http: //www. ifac-control. org

56. International Medical Informatics Association (IMIA)
Add: 16 Place longemalle, CH-1204GENEVA. Switzerland.
Tel: 41-(33) 282649 Fax: 41-(22) 7812322
E-mail: IFIP @ CGEUGE51. bitnet
http: //www. imia. org

57. International Society for Optical Engineering
Add: P. O. Box 10, Bllingham, Washington 98227-0010, U. S. A.
Tel: 1-(360) 676-3290 Fax: 1-(360) 647-1445
http: //www. spie. org

58. Irish Computer Society
Add: Dundrum Castle, Ballinter Rd. DUBLIN 16. Ireland.
Tel: 353-(1) 982692 Fax: 353-(1) 290016

59. Iseland Society for Information Processing
Add: Box 681, 121 REYKJAVIK. Iseland.
Tel: 354-(1) 27577 Fax: 354-(1) 25380

60. John v. Neuman Society for Computing Sciences
Add: P. O. Box 240, Bathori u. 16, H-1368 BUDAPEST. Hungary.
Tel: 36-(1) 1329349 or 1329390 Fax: 36-(1) 1318140
E-mail: huug @ neumann. h

61. Korea Information Science society (KISS)
Add: Room 401, Murijae Bldg. 984-1, Bangbae-3 dong, Seocho-ku, SEOUL 137 063, Rep. of Korea. Korea.
Tel: 82-(2) 5889246 Fax: 82-(2) 5889247
Telex: 22974 qnixsel.

62. Malaysian National Computer Confederation
Add: 46a, Jalan SS2 /66, Selangor Darul Ehsan, 47300 PETALING JAYA. Malaysia.
Tel: 60-(3) 7765160 Fax: 60-(3) 7747026

63. Nederlands Genootschap voor Informatica

Add: Van Diemenstraat 184, NL-1013 CP AMSTERDAM. Nethrlands.

Tel: 31-(20) 6203676　　Fax: 31-(20) 6203669

64. New Zealand Computer Society

Add: P. O. Box 12-249, WELLINGTON. New Zealand.

Tel: 64-(4) 731043　　Fax: 64-(4) 731043

65. Norwegian Computer Society

Add: P. O. Box 6714, Rodelokka, N-0503 OSLO 5. Norway.

Tel: 47-(2) 370213　　Fax: 47-(2) 354669

66. Pattern Recognition Society

Add: 3900 Reservoir Road, N. W. Washington, DC 20007. U. S. A.

67. Philippine Computer Society

成立于 1967 年.

Add: Penthouse c. Rivilla Building, Aguirre Street, Legaspi Viliage, Makati, Metro Manila, Philippines.

Tel: 63-(2) 8180381, 8184227

68. SADIO

Add: URUGUAY 252- 2"D", 1015 BUENOS AIRES. Argentina.

Tel: 54-(1) 405755 or 453950　　Fax: 54-(1) 111877

E-mail: jaiio @ sadio. edu

http: //www. uba. ar /wwws /sadio

69. Singapore Computer Society (SCS)

成立于 1957 年.

Add: 0511, 71 Science Park Drive, NCB Building. Singapore.

Tel: 65-7783901　　Fax: 65-7788221

Telex: NCB RS 38610.

70. Society for Computer Simulation (SCS)

Add: P. O. Box17900, San Diego, CA92177-7900. U. S. A.

Tel: 1-(619) 277-3888

http: //www. scs. org

71. Society for Industrial and Applied Mathematics (SIAM)

成立于 1952 年.

Add: 3600 Univ. City, Science Center, Philadelphia, PA 19104-2688, U. S. A.

http: //www. siam. org

72. Society for Manegement Information Systems (SMIS)

成立于 1968 年.

Add: 111 East, Wacker Drive, Suite 600, Chicago, ILL 60601. U. S. A.
http://www.cba.uga.edu/smis

73. South East Asia Regional Computer Confederation (SEARCC)
Add: c/o National Computer Board, NCB Building, 71 Science Park Drive, SINGAPORE 0511, Singapore.
Tel: 65-7783901 Fax: 65-7788221
Telex: 38610.
http://www.searcc.org

74. SUCESU-NACIONAL
Add: Av. W-3 Norte, Quadra 504, Ed. Mariana, Sala 205, 70730 BRASILIA, Brazil.
Tel: 55-(41) 2256373 (Secr.) Fax: 55-(41) 2282177
Telex: 272161.
http://www.internetional.com.br/sucesu

75. Swedish International Federation for Information Processing(SIFIP)
Add: Box 22114, S-104 22 STOCKOLM. Sweden.
Tel: 46-(8) 6520720 Fax: 46-(8) 6500343

76. Swiss Federation for Information Processing Societies(SVI/FSI)
Add: P. O. Box 71, CH-8037 ZURICH. Switzerland.
Tel: 41-(01) 2757121 Fax: 41-(01) 2727912

77. West African Regional Computer Society (WARCS)
Add: c/o Dr. O. C. Akinyokun, P. O. Box 7600, LAGOS, Nigeria.
Tel: 234-(36) 230747 or 230312
Telex: 34265 cepeda ng.

78. Word Computer Graphics Association (WCGA)
Add: 2033 M St., Suite 399, Washington, DC 20036. U. S. A.
Tel: 1-(202) 715-9556